Identifying Preferred Solutions for Multi-objective Aerodynamic Design Optimization

by

Robert Carrese, B.Eng. (Hons.)

A thesis submitted in fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

School of Aerospace Mechanical and Manufacturing Engineering
RMIT University
Melbourne, Australia

Declaration

Except where due references are made, the work reported in this thesis is solely that of the author alone and has not been submitted or published previously, in whole or in part, to qualify for any other academic award. The content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program, 2nd of March 2009 at RMIT University.

Robert Carrese

Melbourne, June 2012

Abstract

Aerodynamic designers rely on high-fidelity numerical models to approximate, within reasonable accuracy, the flow around complex aerodynamic shapes. The ability to improve the flow field behaviour through shape modifications has led to the use of optimization techniques. A significant challenge to the application of evolutionary algorithms for aerodynamic shape optimization is the often excessive number of expensive computational fluid dynamic evaluations required to identify optimal designs. The computational effort is intensified when considering multiple competing objectives, where a host of trade-off designs are possible. This research focuses on the development of control measures to improve efficiency and incorporate the domain knowledge and experience of the designer to facilitate the optimization process.

A multi-objective particle swarm optimization framework is developed, which incorporates designer preferences to provide further guidance in the search. A reference point is projected on the objective landscape to guide the swarm towards solutions of interest. This point reflects the preferred compromise and is used to focus all computing effort on exploiting a preferred region of the Pareto front. Data mining tools are introduced to statistically extract information from the design space and confirm the relative influence of both variables and objectives to the preferred interests of the designer. The framework is assisted by the construction of time-adaptive Kriging models, for the management of high-fidelity problems restricted by a computational budget. A screening criterion to locally update the Kriging models in promising areas of the design space is developed, which ensures the swarm does not deviate from the preferred search trajectory. The successful integration of these design tools is facilitated through the specification of the reference point, which can ideally be based on an existing or target design.

The over-arching goal of the developmental effort is to reduce the often prohibitive cost of multi-objective design to the level of practical affordability in aerospace problems. The superiority of the proposed framework over more conventional search methods is conclusively demonstrated via a series of experiments and aerodynamic design problems.

Acknowledgments

I would first like to thank my thesis supervisor, Dr. Jon Watmuff, for his invaluable time and support throughout my candidature. I am also very grateful to A/Prof. Hadi Winarto, with whom I developed a very productive working relationship. His expertise and patience have inspired me to complete this research project to the best of my abilities. Considerable thanks also go to Dr. Xiaodong Li, and to Dr. András Sóbester of the University of Southampton, whom with I collaborated extensively. Their guidance and advice have added considerably to my research experience. I am also very grateful to Prof. Sylvester Abanteriba for his continual support.

Throughout my candidature I had the privilege of working with some fantastic colleagues, and am especially grateful to Dr. Manas Khurana. His invaluable (often problematic) experiences have aided me considerably. I am also very grateful to Dr. Upali Wickramasinghe, who first introduced me to the concept of preference-based optimization. Special thanks also go to Lev Lafayette and Craig West from the VPAC labs, and to Samuel Ebenezer from GridPro for kindly offering his assistance. I would also like to thank Viscovery Software GmbH for kindly offering a license for SOMine for the duration of this research. I would also like to thank Tze Min Koh for his suggestions and his willingness to always lend a hand. A big thanks to everyone at the Bundoora campus, particularly Asintha, Sridhar, Xin, Ashwin and Ishan. Special thanks also to the administrative staff, especially Melissa Sigismondo and Lina Bubic. These people and more have made my postgraduate experience as enjoyable (and eventful) as possible.

I am particularly thankful to my family and friends for their continual support and encouragement. Finally I would like to thank God for placing all these people in my path and am deeply grateful to Him for making this research successful and my candidature as enjoyable as possible.

Publications

This thesis report is based on the following publications (or parts thereof):

Carrese, R., and Li, X., 2012. Preference-based multiobjective PSO for airfoil design. *Swarm Intelligence - Introduction and Applications*, Springer (to be published).

Carrese, R., Winarto, H., Li, X., Sóbester, A., and Ebenezer, S., 2012. A comprehensive preference-based optimization framework with application to high-lift aerodynamic design. *Engineering Optimization*. 1–19.

Carrese, R., Sóbester, A., Winarto, H., and Li, X., 2011. Swarm heuristic for identifying preferred solutions in surrogate-based multi-objective engineering design. *AIAA Journal*. 49(7), 1437–1449.

Carrese, R., Winarto, H., Watmuff, J., and Wickramasinghe, U. K., 2011. Benefits of incorporating designer preferences within a multi-objective airfoil design framework. *Journal of Aircraft*. 48(3), 832–844.

Carrese, R., Khurana, M., Winarto, H., and Li, X., 2011. An efficient strategy to incorporate designer-preferences in automated airfoil design. In *the 14th Australian International Aerospace Congress*. Melbourne, Australia.

Carrese, R., Winarto, H., and Li, X., 2011. Integrating user-preference swarm algorithm and surrogate modeling for airfoil design. In *the 49th AIAA Aerospace Sciences Meeting*. Orlando, Florida.

Carrese, R., Winarto, H., and Watmuff, J., 2010. User-preference particle swarm algorithm for airfoil design architecture. In *the 27th International Congress of the Aeronautical Sciences*, Nice, France.

Wickramasinghe, U. K., Carrese, R. and Li, X., 2010. Designing airfoils using a reference point based evolutionary many-objective particle swarm optimization algorithm. In *Proceedings of the Congress of Evolutionary Computation*. Barcelona, Spain.

Contents

List of Figures

List of Tables

List of Symbols

α	Angle of incidence
β	Arbitrary angle
δ	Solution spread
ϵ	Error margin
μ	Mean/Dynamic viscosity
ρ	Air density
σ	Standard deviation
θ	Arbitrary angle
C_l	Coefficient of lift
C_d	Coefficient of drag
C_m	Coefficient of moment
C_p	Coefficient of pressure
d_z	Reference point distance
c	Chord length
f	Objective function
$\hat{f}$	Kriging objective output
K	Archive capacity
L	Length
m	Number of objectives
M	Mach number
n	Number of decision variables
N	Swarm population size
N_K	Kriging dataset size
p	Pressure
$\mathbf{p}_g$	Population global best position vector
$\mathbf{p}_i$	Particle personal best position vector
q	Non-dominated solution

$\mathbf{Q}$	Elitist archive
r	Radius
Re	Reynolds number
$\hat{s}$	Kriging prediction error
t	Time-step
U_∞	Characteristic velocity
$\mathbf{v}$	Particle velocity
x, y, z	Cartesian coordinates
$\mathbf{x}$	Particle position
$\mathbf{X}$	Sampling dataset
$\mathbf{y}$	Fitness output
$\mathbf{Y}$	Kriging response dataset
$\bar{\mathbf{z}}$	Reference point
$\mathbf{z}'$	Preferred solution

Abbreviations

AIAA	American Institute of Aeronautics and Astronautics
CAD	Computational-Aided Design
CFD	Computational Fluid Dynamics
CV	Cross-Validation
DACE	Design and Analysis of Computer Experiments
DTLZ	Deb-Thiele-Laumanns-Zitzler
EMO	Evolutionary Multi-objective Optimization
GA	Genetic Algorithm
GM	Gradient Method
HV	Hyper-Volume
LE	Leading Edge
LHS	Latin Hypercube Sampling
LO	Lower surface
MOP	Multi-Objective Problem
MOO	Multi-Objective Optimization
MOPSO	Multi-Objective Particle Swarm Optimization
NACA	National Advisory Committee for Aeronautics
NASA	National Aeronautics and Space Administration
NSGA	Non-dominated Sorting Genetic Algorithm
PSO	Particle Swarm Optimization
R-NSGA	Reference point Non-dominated Sorting Genetic Algorithm
RANS	Reynolds-Averaged Navier-Stokes
SOM	Self-Organizing Map
TE	Trailing Edge
UP	User Preferences/Upper surface
ZTD	Zitzler-Thiele-Deb

"I have not failed. I've just found 10000 ways that won't work."

- Thomas A. Edison

Chapter 1

Introduction

The continual progress in automated optimization techniques has led to the development of innovative engineering design frameworks. These strategies are applied to the design of a range of aircraft systems and components, and often encompass multiple disciplines and design philosophies. A field which has benefited considerably from advancements in optimization is aerodynamic shape design. The incorporation of modern *evolutionary* optimization for aerodynamic design is however still in its infancy, due to the often excessive number of computational fluid dynamic simulations required to achieve convergence. This poses a more serious problem within a multiple objective environment where a range of trade-off designs are to be identified. The research presented in this thesis aims to further develop the role of evolutionary optimization within the multi-objective aerodynamic design architecture. In this introductory chapter, the research scope and rationale are presented. The outcome of the research is highlighted by identifying the major contributions and findings, which pertain to the concepts introduced in this thesis.

1.1 Motivation

In recent times the use of computational models in engineering design has greatly increased [Keane and Nair, 2005]. Engineers employ high-fidelity numerical models to simulate, within reasonable accuracy, how a complex system behaves. The ability to reflect changes in the behaviour of a system by modifying certain input parameters has driven the use of optimization techniques. This leads to the fundamental question of design:

> *What determines, and more importantly, how can we determine the optimum design to an engineering problem?*

All research in the field of engineering design in some way aims to address this question. From the viewpoint of the designer, the answer to this question is essentially a trade-off

optimization problem – balancing the efficiency of the design process with the fidelity of the numerical models. For this reason, aerospace and aerodynamic design tends to stand at the forefront of optimization development. Simulations may very well be measured in hours and days to achieve the necessary level of accuracy to result in a meaningful design. That is, a design which ensures the anticipated performance is replicated in real flight. This implies an intractable processing time for optimization, and control measures must be implemented to ensure the design process is computationally viable.

1.1.1 Research in Aerospace Design

The insightful text by Keane and Nair [2005] presents a very thorough commentary on the role of computational design in the aerospace industry. A concise roadmap is provided, offering a historical perspective on research in aerospace design to the succession of the computational era. Aerospace design is an immensely broad area, and the research presented in this thesis focuses on the complex area of aerodynamic shape optimization via Computational Fluid Dynamics (CFD). CFD-driven design is a fundamentally difficult task for optimization since the equations that govern fluid flow are highly non-linear and are computationally expensive to solve [Keane and Nair, 2005]. The successful amalgamation of these two disciplines is the key in developing a viable design framework.

Keane and Nair [2005] highlight that the objective of the majority of research in aerodynamic design undertaken within major universities is to improve on the efficiency and sophistication of numerical models. Evidently the motivation here is to improve on modelling capabilities, so that more complex flow phenomena may be accurately studied. Despite the complexity of studying the non-linear mathematics, the outcome of the research is ultimately straightforward, i.e. either the numerical model improves analysis capability or it does not. By comparison, the research undertaken in improving the *synthesis* of aerodynamic optimization frameworks is significantly less. While this level of research may seem less complex, the assessment of a new design process is unequivocally more difficult, i.e. establishing that the design process improves efficiency within an industry setting.

From the physical viewpoint, numerical models have been developed to explore aerodynamic flow phenomena with varying levels of fidelity dependent on the operating regime. From the computational viewpoint, innovative evolutionary optimization techniques have been developed, e.g. by drawing on the Darwinian model of the survival of the fittest or the ability of a flock of birds to move in unison to avoid a predator. While it has been established that evolutionary optimization techniques are the quickest route to identifying globally optimum designs, their use within an aerodynamic design framework is generally

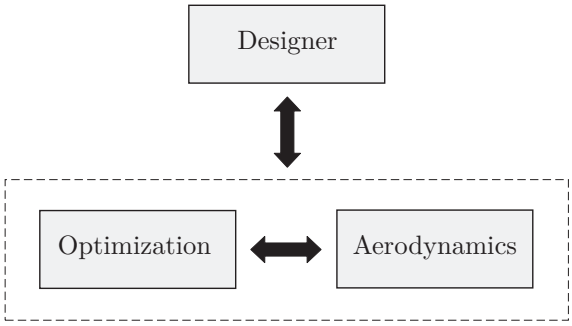


Figure 1.1: The conventional aerodynamic design optimization loop

discouraged [Keane and Nair, 2005]. This unpopularity is due to the often excessive number of simulations required to achieve convergence, thereby restricting the viability of an optimization process. Within a multiple conflicting objective environment, the computational effort is often intensified since there are now any number of trade-off designs which are obtainable. Implementing control measures to improve the efficiency of the design process, without compromising the fidelity, is the dilemma faced by aerodynamic designers. This provides the principal motivation for the current research.

1.1.2 The Role of the Designer

The primary motivation of the research described in this thesis is to improve on the amalgamation of the two disciplines within the design framework. This is essentially the principal role of the designer. It is well known that the success of an optimization process, irrespective of the methods and techniques employed, is ultimately dependent on the experience of the designer in formulating the problem. It may therefore be argued that appropriately defining the role of the designer *within* the design loop is crucial to the efficiency of the optimization process.

A rudimentary example of the conventional aerodynamic design optimization loop is illustrated in Figure 1.1. In this case the designer formulates the problem and allows the optimization loop to run autonomously until the optimum design is identified. While this philosophy is favourable since it requires the least designer input, it may lead to unnecessary computational effort. For example, an autonomous optimization loop may explore certain areas of the design space, which based on the interests of the designer, would otherwise be left unexplored.

An alternative design loop is shown in Figure 1.2. By incorporating the designer within the loop, their domain knowledge may be usefully exploited. It is therefore pos-

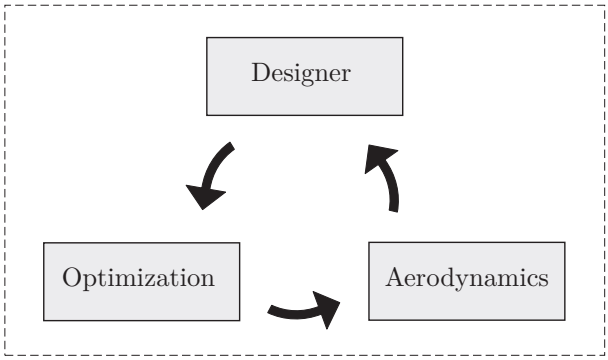


Figure 1.2: The preference-based aerodynamic design optimization loop

sible to focus all computing effort on exploring interesting areas of the design space, to identify only preferred designs best suited to the intended application. Exploiting the designer (or user) preferences to guide the optimization process is not an original concept, and is in fact a fairly popular subject in the literature [see, for example, Fonseca and Fleming, 1998; Coello Coello, 2000; Rachmawati and Srinivasan, 2006; Wickramasinghe, 2010]. Despite the reported benefits, preference-based optimization has yet to make an impact in aerodynamic design. This is a rather remarkable observation, considering that aerodynamic designers could potentially offer significant guidance to the optimizer. The research presented in this thesis attempts to bridge the gap in the synthesis between the designer and the multi-objective aerodynamic design architecture in an attempt to reduce the computational effort required to identify optimal solutions of interest.

1.2 Methodology

The motivation discussed in the previous section has laid a clear foundation from which the research project may be formulated. It is consolidated in the single goal of developing an efficient multi-objective design framework which draws on designer-preferences to identify solutions of interest. The projected outcome of this goal is the practical application of this framework to high-fidelity aerodynamic design problems. This section outlines the scope of the research project, guided by the forming of research questions.

1.2.1 Research Questions

The research described in this thesis is guided by the following questions:

1. **Is there an intuitive method for identifying solutions which reflect the preferred interests of the designer?** This question involves performing a com-

prehensive literature survey on existing methods for multi-objective optimization and preference articulation. The optimization method is responsible for isolating and identifying globally performing regions of a multi-dimensional design space and is the main contributing factor to the efficiency of the design framework. The preference module is the key to identifying solutions of interest to the designer. Simplicity is the key to preference articulation, without compromising automation which is the most desirable feature of a computational design framework.

2. **Can this method be applied to aerodynamic design, such that the domain knowledge of the designer is exploited?** Many methods have been proposed in the literature that may address the question above, yet whether they are applicable to a high-fidelity real-world scenario have not been addressed. Preference articulation is no longer confined to a user stipulating preferred areas of a known design space of a mathematical test problem. The domain knowledge of the designer must be exploited by the optimization framework, yet the designer may have no prior information of the optimization landscape.

3. **Can it be applied to the level of practical affordability for high-fidelity design?** It is of particular concern in engineering design that an optimization process is practically affordable given the often excessive computing effort required to assess every candidate. Control measures should therefore be implemented to ensure that an optimization process is computationally viable. The integration of surrogate modelling and design space visualization techniques are crucial to the success of optimization techniques for high-fidelity design.

1.2.2 Research Scope

The formulated questions provide a very clear roadmap to the successful completion of this research. This is assisted by forming a set of guidelines or milestones which ultimately result in achieving the practical application of this framework for aerodynamic design.

1. **Development of a multi-objective evolutionary algorithm** capable of identifying globally performing areas of the design space.

2. **Implementing the user-preference module** to ideally focus all computing effort on preferred regions of the design landscape, which is a reflection of the preferred interest and experiences of the designer.

3. **Integrating surrogate modelling** to ensure the optimization process is computationally viable for high-fidelity problems of an industry-type setting.

4. **Exploring the use of visualization techniques** to facilitate the optimization process and extract useful information from the optimization landscape.

5. **The successful integration of these design tools** and the synthesis of the framework for aerodynamic design applications.

This thesis report is organized such that these milestones are discussed sequentially, highlighting to the reader the benefits of adopting each design tool. The efficiency of the developed framework is illustrated through its application to aerodynamic design problems of varying complexity and fidelity.

1.2.3 Contributions

This research makes novel contributions in the fields of preference-based optimization and visualization, as well as improving the synthesis of the aerodynamic design architecture. These contributions originate from the development and integration of the user-preference module, which is tailored to reflect the preferred interests of the designer. This section describes some of the key novel contributions of this research:

- The multi-objective particle swarm optimization algorithm developed in this thesis is quick and simple to implement, and it provides a superior convergence rate over other multi-objective methods. The concept of utilizing a secondary population in the form of an archive allows identified non-dominated solutions to be stored and subsequently used to further guide the search. The Gaussian mutation operator applied to members in the archive is ideal for the management of highly multi-modal problems.

- The user-preference module presented in this thesis takes the form of a reference point. This provides additional guidance to the optimizer to converge to the preferred region of the Pareto front, thereby focusing all computing effort on identifying solutions of interest. The novel contribution is the integration of the reference point module to an aerodynamic design framework. In this case the reference point is tailored to reflect a target or existing design, and thus is used to place performance priority on specific design conditions. A procedure to control the location of the preferred region is introduced, by emphasizing or prioritizing certain design conditions in order to identify designs which best suit the intended application.

- A primary novel contribution of this research is utilizing the preference-based architecture to manage high-fidelity aerodynamic design problems with the aid of sur-

rogate modelling. The benefits of preference-based optimization have not made an impact within high-fidelity engineering design, thus not driving any research in this area. The introduction of the Kriging component allows for the management of high-fidelity problems restricted by a computational budget. The search effort remains consistent, with the added advantage that less precise evaluations are performed.

- An active area of research in surrogate-based optimization is the development of screening criteria which determine, with sufficient confidence, which candidate designs are promising for precise evaluation. In this research, a novel screening criterion is developed which utilizes the information provided by the reference point to screen candidates. Screening criteria are to be implemented with caution, so that a fair balance between searching less explored regions of the design space and exploitation of promising areas can be established. This simple, yet logical, criterion is proficient in identifying solutions which are expected to provide improvement within the preferred region of the design space. The novel reference point criterion utilizes the adaptive control and lower-bound confidence strategies to identify solutions in the population which are feasible for precise evaluation. This strategy effectively maintains control over the number of identified solutions as the swarm begins to exploit the preferred region.

- Another major contribution is the practical application of design space visualization and data mining techniques. Such techniques are applied to extract useful information on the relationship between the design space and the objective space. However when confronted with multiple conflicting objectives, related studies have demonstrated that extracting case-specific information can be challenging. The introduction of the reference point allows candidate designs (which are characterized by multiple objective values) to be assessed as a single scalar. Hence certain variables or objectives which directly drive the optimization process (rather than a specific design condition) are easily identified. Visualization via the reference point metric therefore facilitates the interpretation of a multi-objective optimization environment such that meaningful conclusions can be made during the pre-optimization and decision making processes.

- The developmental effort and successful integration of all the design tools described in this thesis is facilitated through the specification of the reference point. Hence the synthesis of the aerodynamic design architecture is improved by incorporating the domain knowledge and preferred interests of the designer. The phases of optimization are all linked via the reference point distance metric, the value of which is a measure

of the resemblance to the preferred interests of the designer. Hence conclusions regarding the dimensionality of the design space (since variables are now categorized on how they drive the design rather than their effect on specific objectives), the computational budget (since the design space is effectively reduced to encompass only the preferred region), or selecting the most appropriate design (since final designs are characterized via their resemblance to the reference point compromise) can now be expertly made, rather than relying on estimations.

1.3 Thesis Outline

This thesis is comprised of eight chapters and an appendix. Chapter 2 presents a concise literature review of the elements of the aerodynamic design architecture, with an introduction to optimization theory and techniques provided in Chapter 3. A brief introduction to surrogate modelling and its management within an evolutionary optimization framework is given in Chapter 4. The developmental effort of the preference-based design framework is documented in Chapters 5 and 6. Results of the application of the design framework to aerodynamic problems is provided in Chapter 7. The final chapter summarizes the outcomes of this research and offers recommendations for potential areas warranting further research and development.

Chapter 2

Elements of Aerodynamic Design

Aerodynamic design originates from an understanding of the fundamental physics of flight. It has evolved from the use of wind tunnel catalogues and *cut-and-try* methods, to modern computational frameworks. Since the introduction of computational methods researchers have strived to increase modelling capabilities to approximate the flowfield around aerodynamic shapes. While such methods allow for the study of aerodynamic flow phenomena, they have also become an integral component of computational aerodynamic design. This chapter provides a short description of the advancements in computational aerodynamic design and numerical modelling, and discusses the physical elements of the aerodynamic design architecture in detail.

2.1 Aerodynamic Design Architecture

The aim of implementing an aerodynamic design process is to identify or conform to the best possible shape for the given design requirements. The introduction of computational frameworks further aims to increase the level of automation [Labrujère and Sloof, 1993]. While automation simplifies the design cycle, success is still largely dependent on the fidelity of the computational methods, as well as the experience of the designer in formulating the design problem. This section provides a short description of two design strategies reported in the literature, namely the inverse and direct approaches. For a more detailed discussion, the reader is referred to the paper by Labrujère and Sloof [1993].

2.1.1 Inverse Methods

Early attempts at developing computational methods for airfoil and wing design focused on solving the inverse problem. The inverse design methods originate from the method of conformal mapping of Lighthill [1945], which also provided the foundation for the full

potential hodograph methods [see Drela and Giles, 1987]. The inverse method involves specifying the appropriate design requirements in terms of a target pressure distribution. The aerodynamic shape is then iteratively conformed to provide the prescribed surface pressure distribution. The advantages of this method are fairly evident, since the designer has the flexibility to control local flow behaviour. For example, extended regions of laminar flow can translate to extending the favourable pressure gradient on the upper surface, or the aft loading can be controlled by specifying the surface pressure difference. Despite the apparent theoretical advantages, the inverse design method is fairly limited. In both the studies of Labrujère and Sloof [1993] and Yiu [1994] it is confirmed that specifying a target pressure distribution and ensuring a feasible airfoil profile exists is a major design challenge. Furthermore, appropriate geometrical constraints must be implemented in order to avoid a self-intersecting profile, or to adhere to minimum thickness requirements. Labrujère and Sloof further conclude that revising the target pressure distribution is inevitable in order to obtain a manufacturable profile. The development of more robust design strategies addresses the demerits of the inverse design method.

2.1.2 Direct Methods

Direct methods refer to the philosophy of using mathematical optimization methods in design to identify the optimal shape that achieves the prescribed design criteria. The generalized framework for an aerodynamic shape optimization process is demonstrated in Figure 2.1. The use of optimization methods in aerodynamic design frameworks is increasing [Keane and Nair, 2005] and the direct approach is essentially applicable to any aerodynamic cost function [Labrujère and Sloof, 1993]. Furthermore, this strategy allows for the explicit treatment of multiple objective functions over a wide operational spectrum. In principle, the computational flow solver is viewed as a black-box used to calculate the objective function(s). The integrated optimization algorithm iteratively determines the necessary shape modifications to a candidate shape (for population-based optimization methods there are multiple candidate geometries, initialized randomly in the design space) by direct manipulation of the shape design variables, to achieve the objective(s). Convergence is achieved once the optimal solution (or an approximate) has been obtained, or the computational budget is exceeded.

As highlighted in the flowchart shown in Figure 2.1, the success of the direct approach is dependent on three main components within the design loop; the shape parameterization method, the computational flow solver and the optimization algorithm. All aerodynamic design strategies share a common requirement that the geometry is represented by a finite number of design variables. A method to mathematically parameterize shapes is therefore

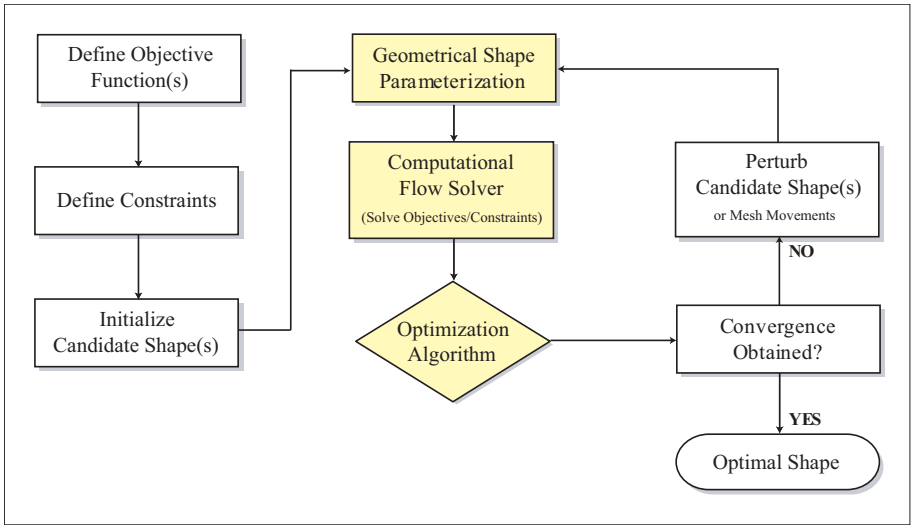


Figure 2.1: Generalized process flowchart for direct aerodynamic shape optimization

required, so shape modifications can be made via direct manipulation of the design variables. The number of design variables is directly proportional to the geometrical degrees of freedom and therefore governs the dimensionality of the problem. The computational flow solver is used to determine the flowfield properties surrounding the aerodynamic shape. The objective function is obtained from the flow solver and may represent the global performance of the shape via aerodynamic coefficients. It is therefore up to the discretion of the designer to appropriately formulate the objective and constraint functions, such that they reflect the design and operating requirements. The flow solver has a significant influence on the efficiency of the optimization process since repeated evaluations of the objective function are required for each candidate shape. Furthermore, the accuracy of the flow solver ultimately governs the overall fidelity of the computational aerodynamic design method. The responsibility of the optimization algorithm is to iteratively determine the shape modifications required to satisfy the objective, whilst adhering to any shape or performance constraints. The optimizer should be robust and applicable to a wide operational spectrum, yet efficient to guarantee convergence with the least computational expense. A review of optimization concepts and techniques is provided in Chapter 3.

The direct approach originated from one of the first published accounts of wing aerodynamic design via numerical optimization by Hicks and Henne [1978]. In this work, a full potential equation solver and analytical parameterization method was combined with a conjugate gradient optimizer for transonic wing design. Since then, elaborate design methodologies have been developed, owing also to the work of Jameson [1988] who developed the use of control theory via the solution of adjoint problems, and applied it to design using the Navier-Stokes equations [Jameson *et al.*, 1998]. A review of aerodynamic

design frameworks reported in the literature will be provided in later sections, as concepts are further introduced. Each of these reported frameworks aims to address the shortcomings of the direct numerical approach, largely a result of the increased search effort to identify globally optimal shapes using high-fidelity computational methods. Despite the advancements in computer capabilities, the use of the direct approach for numerical optimization is still a computationally challenging and intensive undertaking. The extension to multiple objectives leads to a more generalized problem formulation, which significantly increases the computational cost of convergence. This provides the principal motivation for the aerodynamic design framework described in this thesis, where control measures are proposed to ensure an efficient framework, focusing on identifying solutions to suit the preferences of the designer.

2.2 Geometrical Shape Parameterization

Geometry manipulation is of particular importance in aerodynamic design. The pressure for automated approaches has led to the advent of Computer-Aided Design (CAD) packages, which are now seen as the fundamental tool for aerospace design [Keane and Nair, 2005]. With the evolution of CAD, designers have always sought to integrate some of the more higher-order parameterization methods for rapid design changes. These methods may have more generic capabilities (e.g. splines) or may be devoted to a specific class of shapes (e.g. airfoils, fuselage cross-sections, etc.). In aerodynamic design, the selection of the shape parameterization method is an important contributing factor since it will effectively define the objective landscape and the topology of the design space [Song and Keane, 2004]. Furthermore, certain parameterization techniques are tailored for specific forms of optimization. Aside from reducing the computational cost to generate feasible shapes, there are a number of criteria which are essential for geometry parameterization [Jin and Sendhoff, 2009]. In this research, the criteria outlined by Sóbester [2009] are adopted for the selection and application of shape parameterization methods:

- **Conciseness:** The number of shape variables governs the dimensionality of the design space. The size of an n-dimensional space observed at q levels is $\mathcal{O}(q^n)$ which implies an exponential growth commonly referred to as the *curse of dimensionality*. Therefore to reduce the computational cost of a conventional design process, it is paramount to explore possibilities of reducing the number of design variables.

- **Robustness:** This feature reflects the ability of a parameterization method to yield physically and geometrically sensible shapes, across the entire design space. A low

degree of robustness results in an inefficient optimization process, due to infeasible candidates. Obviously, one may be rigorous in restricting the design variable range as to reduce the probability of generating infeasible shapes. However this may result in completely neglecting high performing regions of the design space.

- **Flexibility:** This refers to the breadth of shapes that a parameterization method is capable of exploring. The degree of flexibility is difficult to measure, since there is no means of determining whether a method is exploring a sufficient range of shape classes. Innovation in shape design is generally a result of exploring shapes which perhaps are not immediately intuitive or conventional, and is only possible by increasing the flexibility of the parameterization method.

If the aim of the optimization process is to improve on an established design, then perhaps local parameterization methods are desirable since there are a greater number of geometrical degrees of freedom. However, the large number of variables may prove to deteriorate the convergence rate for global design applications. The development of efficient parameterization models has therefore been given significant attention, with the aim to maximize the flexibility of geometrical control with a minimum number of design variables. The classic NACA series as described by Abbot and von Doenhoff [1959] was one of the earliest attempts at defining an airfoil parameterization method for design. This method provides a wide range of airfoil shapes, by controlling important aerodynamic features such as thickness and camber. While these methods are rarely adopted for industrial applications, they have inspired the more higher-order methods developed by researchers [Keane and Nair, 2005]. A comprehensive survey of shape parameterization methods for numerical optimization is given by Samareh [2001] – In this thesis the discussion is limited to the popular methods used in aerodynamic optimization frameworks.

2.2.1 Discrete Method

The simplest form of manipulating geometry is to directly use the the grid-point coordinates as design variables. The main advantage of this method is that it can be applied to almost any aerodynamic shape, and allows for a flexibility which is only restricted by the number of design variables. While this method may be suitable for local parameterization, it is not viable for global shape design [Samareh, 2001]. Since the geometry is manipulated through grid-point translation, a smooth geometry is difficult to maintain and the point-wise gradients are not continuous [Braibant and Fleury, 1984; Samareh, 2001]. Furthermore, constraints must be imposed on all grid-point translations, to ensure a feasible shape is generated (e.g. self-intersecting profiles). Methods to overcome the

demerits of the discrete approach have been investigated, for example in Jameson *et al.* [1998], where the gradient with respect to the grid points was conditioned to maintain smooth geometries. The fact still remains however that for global design with a large number of grid-points, successful convergence is improbable, and the computational cost is quite high [Samareh, 2001].

2.2.2 Spline-based Methods

The discrete method allows for greater flexibility in geometry manipulation, yet a global optimization process is heavily restricted by the geometrical degrees of freedom and hence the number of design variables. The desire to allow for global geometrical control, whilst controlling local curvature and continuity has therefore led to the introduction of the spline-based methods [De Boor, 1994]. When used effectively, spline-based methods can maintain a similar level of flexibility to discrete methods, whilst the number of design variables is significantly reduced. This suggests that the spline-based methods are a viable option for aerodynamic optimization [Braibant and Fleury, 1984]. Curve-fitting was introduced in the NACA airfoil series, which used quadratic polynomials to model the thickness and camber distributions [Abbot and von Doenhoff, 1959]. Cubic interpolation methods have also been reported in the literature. For example, Marsden *et al.* [2004] reports using Hermite interpolation for aeroacoustic shape design of a beveled trailing edge. The advantage of this method is that curves are shape-preserving, therefore geometrical constraints are easily enforced. It was demonstrated that by increasing the number of control nodes, there is greater geometrical flexibility yielding improved designs.

The Bézier method is classical curve-fitting, where control points and Bernstein polynomials are used to define individual curves [Samareh, 2001]. The control points form a polygon and are generally treated as design variables [Samareh, 2001]. The Bézier form is effective for simple curves, however it parameterizes the shape globally and perturbing one control point subsequently modifies the entire shape contour. It is therefore not advised to model high-degree Bézier curves for shape optimization [Samareh, 2001]. For complex curves, a composite of several low-degree Bézier curves can be used to model the entire curve, termed the B-spline approach. Shape parameterization via B-splines has been reported in the literature for aerodynamic design. Anderson and Venkatakrishnan [1997] report using the B-spline parameterization approach in conjunction with an unstructured flow solver for a number of aerodynamic design test cases. Jones *et al.* [1998] also reports the use of B-splines for an aerodynamic and aeroacoustic optimization design case-study. B-splines are also used in the aerodynamic design framework of Nemec and Zingg [2002] as well as their subsequent work in multi-point optimization [Nemec *et al.*, 2004]. Song

and Keane [2004] compared the use of B-splines and orthogonal basis functions for various airfoil shapes. It was concluded that B-splines are considerably more accurate at representing airfoils. However this is at the expense of a higher computational cost.

B-splines have proven to be an established method for shape parameterization, yet its main deficiency is its inability to accurately represent conic sections [Samareh, 2001; Keane and Nair, 2005]. Furthermore since it is inherently a local parameterization method, a greater number of design variables are required to maintain global control. To address the deficiencies of both the Bézier and B-spline approaches the Non-Uniform Rational B-Splines (NURBS) are introduced, which are the most general form of spline curve Piegel and Tiller [1996]. This method is a direct extension of the B-spline approach, with the introduction of weight terms for each control point. The weights affect the magnitude of manipulation of each independent control point, which is beneficial for both local and global control. NURBS are the basic representation tool of most CAD systems and are used to represent most of the shape classes needed for aerospace design [Samareh, 2001].

While NURBS is favourably applicable to all forms of geometry, it presents some design issues for aerodynamic shape optimization. There are no set guidelines for determining the weight terms, and ill-definition of the weights may eventually lead to sub-optimal or irregular geometries. Lèpine *et al.* [2001] and Painchaud-Ouellet *et al.* [2006] both report the use of the NURBS for aerodynamic wing design. The studies demonstrate that a large number of design variables are required for accurate geometrical representation. Keane and Nair [2005] also do not suggest the use of NURBS to model complex aerodynamic shapes such as transonic airfoils, as they require a large number of control points. Furthermore, as highlighted by Kulfan [2008], there is no intuitive relationship between the aerodynamics and the design variables, which the optimizer could exploit. For this reason, wherever possible, it is beneficial to adopt a parameterization scheme which allows strict control over important aerodynamic features of the geometry.

2.2.3 The PARSEC Method

Thus far the aerodynamic shape parameterization tools that have been described are quite general and can be used to parameterize an arbitrary geometry. For certain applications, it is possible to make use of fundamental aerodynamic theory to refine the parameterization method, such that the design variables relate to important aerodynamic or geometric quantities. A common method for airfoil shape parameterization is the PARSEC method by Sobjieczky [1998]. It has the advantage of strict control over important aerodynamic features, and it allows independent control over the airfoil geometry for im-

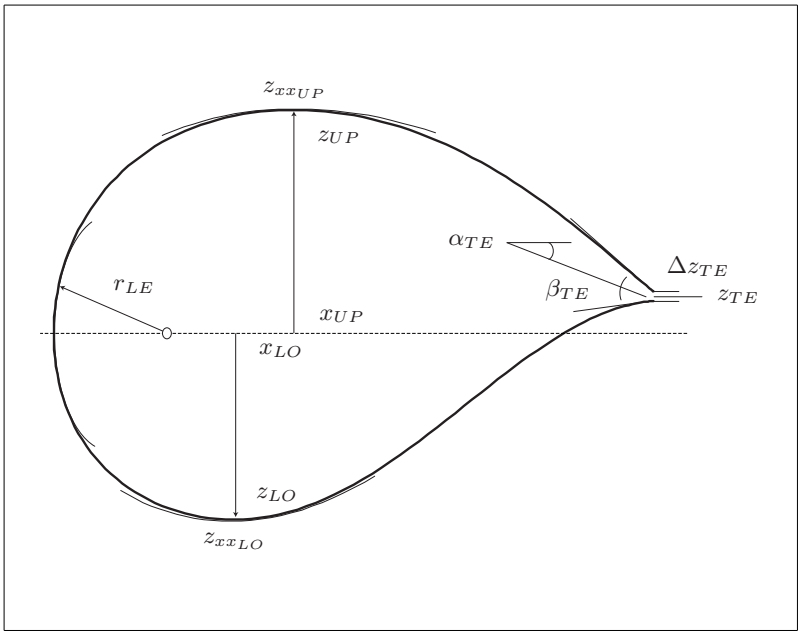


Figure 2.2: Airfoil representation via the PARSEC method

posing shape constraints. The methodology is characterized by eleven design variables (see Figure 2.2) which are the leading edge radius (r_{LE}), upper and lower thickness locations ($x_{UP}, z_{UP}, x_{LO}, z_{LO}$) and curvatures ($z_{xx_{UP}}, z_{xx_{LO}}$), trailing edge direction (α_{TE}) and wedge angle (β_{TE}), trailing edge coordinate (z_{TE}) and thickness (Δz_{TE}). The shape function is modelled via a sixth-order polynomial function:

$$z_k = \sum_{n=1}^{6} a_{n,k} \cdot x_k^{n-\frac{1}{2}} \tag{2.1}$$

where (x, z) are the shape coordinates and k denotes either the upper (suction) or lower (pressure) airfoil surface. The coefficients a_n are determined from the geometric parameters. A complete description on obtaining the polynomial coefficients is given by Castonguay and Nadarajah [2007]. The PARSEC method has been extensively applied for aerodynamic optimization in the literature. Ray and Tsai [2004] report the use of the PARSEC method for transonic airfoil optimization. In this study the design space is reduced to nine dimensions as the parameters z_{TE} and Δz_{TE} were held constant. Jeong *et al.* [2005b] also use the PARSEC method in their data mining analysis of a transonic aerodynamic design space. Quantitative information via the decomposition of variance, and qualitative information via self-organizing maps was obtained to establish the relationship between the PARSEC variables and the objectives. The non-linear sensitivity of the transonic landscape to airfoil thickness was established.

In the response surface based optimization scheme of Vavelle and Qin [2007], the PARSEC method is utilized to optimize the outer wing profile of a blended wing body

configuration. Since the PARSEC method provides direct control over the airfoil thickness, constraints on the minimum thickness-to-chord ratio were easily enforced by specifying the variable limits. This was also the case in the transonic airfoil design framework of Carrese *et al.* [2011b], where the thickness was constrained by the upper and lower limits of z_{LO} and z_{UP} respectively. It was also concluded that the pressure surface thickness is of significant importance, which confirms the study of Jeong *et al.* [2005b]. In the airfoil shape optimization framework of Khurana *et al.* [2008b; 2010], data mining tools were applied to quantify the relationship between the PARSEC variables and their effect on subsonic airfoil geometry and aerodynamics. The experiments confirm the one-to-one geometric control of the PARSEC variables. The influence of the upper thickness location on the drag coefficient was highlighted, as this parameter directly influences the boundary layer transition point. Furthermore, the influence of the leading edge radius and thickness to the maximum lift coefficient was confirmed. The variable Δz_{TE} was eliminated as it yielded a relatively negligible influence to the aerodynamic design space.

Although the original PARSEC method is flexible enough to generate a wide spectrum of airfoil families, certain modifications have been proposed in the literature tailored for specific operating conditions. For supercritical transonic airfoils, Sobjieczky [1998] suggested additional control over the trailing edge curvature, which is beneficial to reduce the probability of downstream boundary layer separation, giving rise to increased drag. The modification allows for divergent trailing edge modelling, which provides a concave surface shaping with curvature continuously increasing towards the trailing edge. A new variable $\Delta\alpha_{TE}$ was introduced, which directly influenced the additional curvature of the trailing edge. The addition of bump functions was also proposed for local curvature control. This allows the geometry to be modified at local supersonic regions to eliminate shocks. A modification to the original PARSEC method for subsonic airfoil representation was suggested and implemented by Khurana and Winarto [2010]. The modification addresses the inability of the original PARSEC method to generate airfoils with a highly cambered nose. The generation of the airfoil shape is decoupled with upper and lower airfoil contours parameterized independently. Because of the introduction of three additional design variables, the modified PARSEC method performed favourably for the shape optimization study compared to the original method.

The modification for additional trailing edge curvature by Sobjieczky was later extended by Jahangirian and Shahrokhi [2009]. It was confirmed that the original PARSEC method does not provide sufficient geometrical flexibility at the trailing edge to conform to certain supercritical profiles. Despite the introduction of the Sobjieczky variant, it

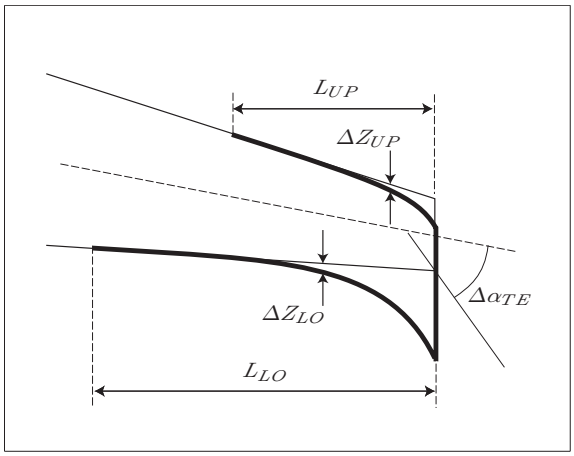


Figure 2.3: Additional trailing edge curvature via the modified PARSEC method

is still not a viable approach as it may lead to self-intersecting geometry, as shown by Jahangirian and Shahrokhi. The new modification decouples the trailing edge parameterization by first defining a smoother upper surface contour and then constraining the lower surface to intersect the trailing edge coordinate. Figure 2.3 illustrates the modification to the trailing edge curvature proposed by Jahangirian and Shahrokhi.

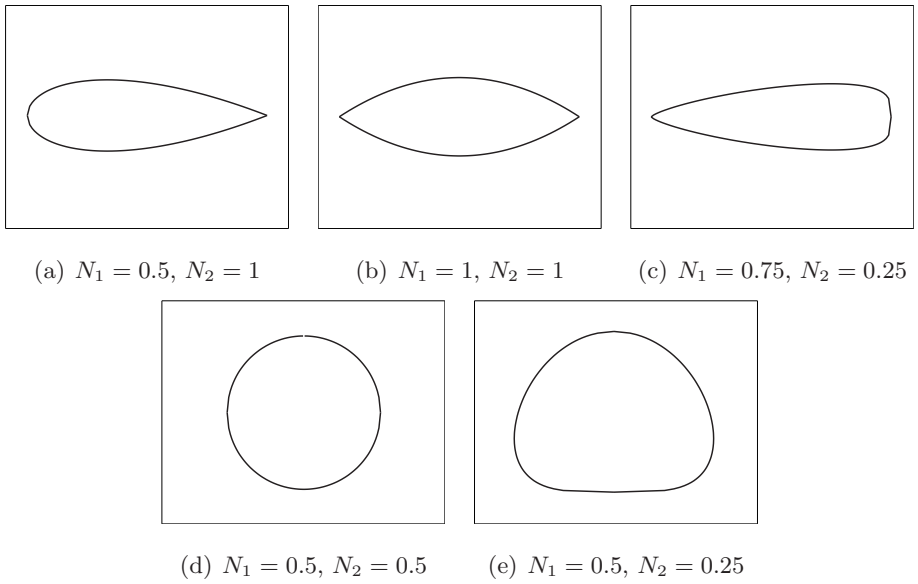
The trailing edge modification is applied to the upper and lower surfaces as follows:

$$\delta_z = \frac{L \cdot \tan \Delta\alpha_{TE}}{2\mu\tau} \left[1 + \eta \cdot x^{\tau} - (1 - x^{\tau})^{\mu}\right], \tag{2.2}$$

where the constants η, μ, and τ are set to 0.8, 2, and 6 respectively. The modification is applied over the entire surface, such that $L_{UP} = L_{LO} = c$ where c is the airfoil chord length. The modification by Jahangirian and Shahrokhi provides a smoother upper surface contour with respect to the original modification of Sobieczky, which is favourable to reduce the pressure drag of the upper surface. The performance of this model is compared to the Sobjieczky modification as well as the original PARSEC method through a series of geometrical and aerodynamic inverse design studies. It is demonstrated that the new model is capable of conforming more accurately to the desired profile, by either specification of the target geometry or the target pressure coefficient. Furthermore Jahangirian and Shahrokhi have shown that convergence is obtained at a reduced computational cost.

2.2.4 The CST Parametric Method

The PARSEC parameterization method is flexible enough to conform to many vastly different airfoil profiles. However the demerit of this method is that it is generally not applicable to other aerodynamic shapes, limiting its application to airfoil shape optimization. A global parameterization technique, capable of modelling arbitrary geometry of



(a) $N_1 = 0.5$, $N_2 = 1$ (b) $N_1 = 1$, $N_2 = 1$ (c) $N_1 = 0.75$, $N_2 = 0.25$

(d) $N_1 = 0.5$, $N_2 = 0.5$ (e) $N_1 = 0.5$, $N_2 = 0.25$

Figure 2.4: Illustration of aerodynamic shapes via class function representation: (a) Airfoil; (b) Ellipsoid; (c) Projectile; (d) Circular; and (e) Fuselage cross-section

varying curvature was developed by Kulfan and Bussoletti [2006] and is known as the method of Class function/Shape function Transformation (CST). Apart from its ability to parameterize round nose and sharp aft end geometries along the streamwise axis, the CST method can also be tailored to represent cross-sections of fuselages, channels and ducts. The mathematical description of the geometry is represented through a combination of a shape function and class function. The class function provides a global representation of the desired geometry and is of the form:

$$C(x) = (x)^{N_1} \cdot (1 - x)^{N_2}, \tag{2.3}$$

where the exponents N_1 and N_2 define a variety of basic general classes of geometries such as airfoils, ellipsoids, elliptic bodies, biconvex bodies, etc. as highlighted in Kulfan [2008]. Figure 2.4 illustrates several examples of generic shapes that are possible by varying the exponents N_1 and N_2. Shape functions are used to conform to a specific shape by perturbing the generic shape. Kulfan provides an example of an airfoil shape function, where geometrical quantities such as leading edge radius, trailing edge angles and thickness distribution are specified. It is also shown that by specifying shape functions through Bernstein polynomials of varying order, any smooth airfoil may be generated.

The CST method has been successfully applied in two- and three-dimensional wing optimization studies. CST was utilized in the transonic wing shape optimization framework of Bogue and Crist [2006], where it was concluded that a Bernstein polynomial order of six or higher was required to provide consistent results. The performance of the CST method

was compared to the discrete and B-spline parameterization techniques by Mousavi *et al.* [2007]. In the inverse design case-study it was concluded that a by increasing the order of the Bernstein polynomials, the target pressure is obtained with reasonable accuracy. For the three-dimensional shape optimization study, Mousavi *et al.* demonstrated that the CST method was unable to attain the same level of drag reduction compared to the other parameterization techniques. While these studies focus on airfoil and wing applications, the full potential of the CST method lies in its ability to represent a wide range of shape classes of interest to aerodynamic design. The generic class and unit shape functions may be varied in order to conform to a specific family of shapes. This is evident from Carrese *et al.* [2011c], where the class function of Kulfan was used as a baseline to generate a large breadth of double-lobe fuselage shapes.

2.3 Computational Flow Solver

Characterizing a candidate shape is achieved through definition of the objective and constraint functions, which are obtained from the flow solver. The efficiency of the optimization framework is dependent on the selection of the flow solver, since it is the most computationally expensive component and repeated evaluations of the objective and constraint functions are required for each candidate shape. However if the flow solver is not sufficiently accurate, the optimization process will converge to shapes which exploit the numerical errors or limitations, rather than the fundamental physics of the problem. For this reason, it is desirable to maintain the correct balance between solution accuracy and computational expense, which is dictated by the flow regime. For certain problems where the aerodynamic flowfield is well behaved, it may be sufficient to consider more robust linear solvers. However for high-fidelity design it is prudent to consider non-linear and more computationally demanding solvers, to ensure optimized shapes provide the anticipated performance requirements in flight.

2.3.1 The Navier-Stokes Equations

The governing equations of motion for fluid flow are the Navier-Stokes equations. The Navier-Stokes equations are highly non-linear and thus no general analytical solution exists. The advent of computational and numerical modelling has however given rise to a new discipline for solving aerodynamic problems - Computational Fluid Dynamics (CFD). The distinguishing feature of CFD is that the flow-field must first be divided into a number of discrete points or nodes. Coordinate lines are traced through each node to generate the grid. The flow-field properties are then numerically calculated at discrete points in space

and time. The Navier-Stokes equations are derived from the principles of the conservation of mass, momentum and energy. These equations may be derived using infinitesimal or finite control volume approaches and are expressed in either differential or integral form. A short description of the Navier-Stokes and other related equations is presented here. A thorough discussion on these equations is given elsewhere [Anderson *et al.*, 1984; Thompson, 1986]. The unsteady, compressible form of the Navier-Stokes, continuity and energy equations are expressed in the two-dimensional domain as:

- Conservation of momentum (Navier-Stokes)

$$\frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho u^2)}{\partial x} + \frac{\partial(\rho uv)}{\partial y} = -\frac{\partial p}{\partial x} + \frac{1}{Re}\left[\frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{xy}}{\partial y}\right], \tag{2.4}$$

$$\frac{\partial(\rho v)}{\partial t} + \frac{\partial(\rho uv)}{\partial x} + \frac{\partial(\rho v^2)}{\partial y} = -\frac{\partial p}{\partial y} + \frac{1}{Re}\left[\frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \tau_{yy}}{\partial y}\right]. \tag{2.5}$$

- Conservation of mass (continuity)

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y}. \tag{2.6}$$

- Conservation of energy

$$\frac{\partial E_T}{\partial t} + \frac{\partial(uE_T)}{\partial x} + \frac{\partial(vE_T)}{\partial y} = -\frac{\partial(up)}{\partial x} - \frac{\partial(vp)}{\partial y} - \frac{1}{RePr}\left[\frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y}\right] + \frac{1}{Re}\left[\frac{\partial}{\partial x}(u\tau_{xx} + v\tau_{xy}) + \frac{\partial}{\partial y}(u\tau_{xy} + v\tau_{yy})\right]. \tag{2.7}$$

Where p, ρ, Re and E_T denote the pressure, density, Reynolds number and total energy respectively. The variables u and v represent the components of velocity in the spatial coordinates x and y respectively and the time is denoted as t. The variables τ are components of the stress tensor, which express the viscous stresses in terms of velocity variations. The conservation of energy is technically distinct from the Navier-Stokes equations, yet it is important for compressible flowfields, and introduces the heat flux q and the Prandtl number Pr which relates the viscous to thermal stresses. The inertial components of the momentum equation are known as the convection terms which dictate the space position of the flow particle. The viscous components are termed the diffusion terms.

2.3.2 Quantities in Aerodynamic Design

Of particular interest in aerodynamic design are bodies for which the force parallel and opposite to the direction of motion (drag) is significantly smaller in magnitude to the force component acting normal to the direction of motion (lift). The pressure difference of the upper and lower surface produces the resultant lift force. The location where the

resultant lift force acts generates a moment, which governs the magnitude and direction of the pitch inclination. Drag is comprised of both pressure and viscous components, the latter dominating in subsonic flight. Viscosity of the flow in the immediate vicinity of the surface (boundary layer) generates surface shear stress from which drag is derived. For the given operating condition, deviations to the shape geometry will result in variations to the pressure and shear stress distributions, which in turn alter the force values. In aerodynamic design and analysis, it is customary to express these forces as non-dimensional coefficients. It follows, that for a geometrically similar airfoil at a given flow incidence angle (α), the lift (C_l), moment (C_m) and drag (C_d) coefficients are a function of:

$$[C_l,\ C_m,\ C_d] = f\left(Re, M\right). \tag{2.8}$$

The Reynolds number Re is the dimensionless ratio of the inertial forces to viscous forces and quantifies their respective relevance for a given operating condition, such that

$$Re = \frac{\rho U_\infty c}{\mu}, \tag{2.9}$$

where U_∞ and c are the characteristic (freestream) velocity and length of the problem, and μ is the dynamic viscosity. It is observed that as $Re \to \infty$, the diffusion terms of the momentum equation can be neglected and the problem can be characterized purely by the generation of inertial forces. The Mach number M is another dimensionless parameter which is applicable to compressible flows, and is the ratio of the characteristic velocity to the freestream speed of sound a_∞, such that

$$M = \frac{U_\infty}{a_\infty}. \tag{2.10}$$

The Navier-Stokes equations in their complete form are highly nonlinear. Dependent on the flow regime (i.e. Re and M), simplifications are made to achieve a practical solution. Various physical simplifications can be made resulting in varying levels of fidelity. For example, the viscosity of the fluid can be neglected or confined to a thin shear layer over the aerodynamic surface as per the Prandtl boundary layer concept [Schlichting *et al.*, 2004]. A description of the CFD techniques utilized in this research is presented.

2.3.3 Panel Methods

The boundary element or panel methods are used to describe inviscid, irrotational and incompressible flow. The level of assumptions restrict the domain of application of the panel methods, however within the appropriate domain they are very advantageous. Panel methods require only surface discretization. Panel methods compute the inviscid pressure distribution about arbitrary lifting bodies by solving the Laplace equation via flow singularities. The inviscid pressure distributions are generally iteratively coupled to empirical

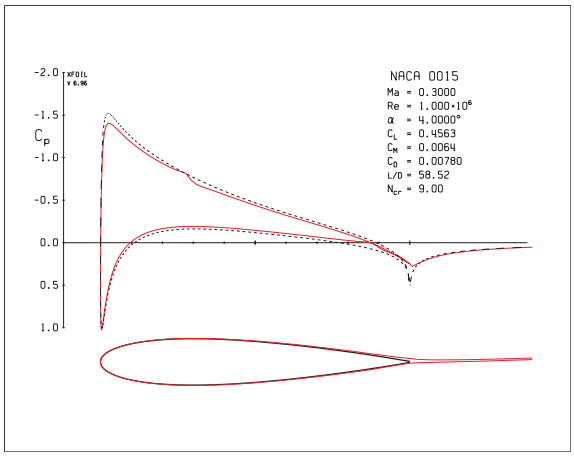


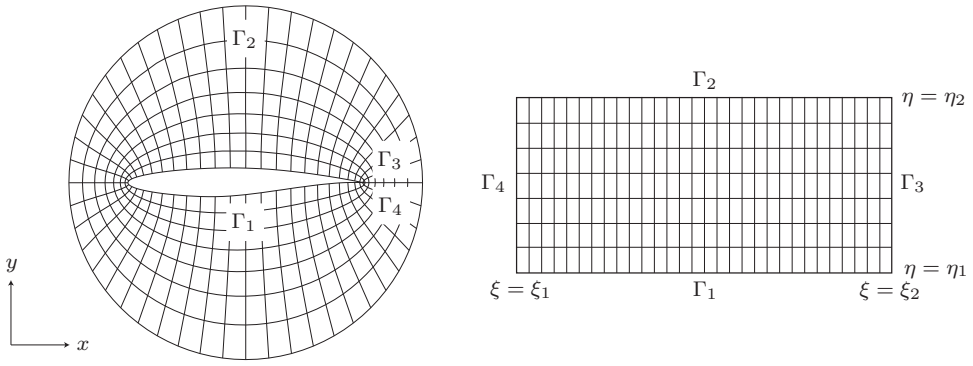
Figure 2.5: XFOIL simulation of NACA0015 airfoil at angle of incidence

boundary layer models to measure viscous effects. The panel methods are still in use for industrial applications, with three-dimensional codes such as VSAERO. Another widely used panel method code is XFOIL developed by Drela [see Drela and Giles, 1987]. XFOIL is a subsonic airfoil analysis code, which superimposes uniform flow and vortices acting at the airfoil surface to simulate the surrounding flow-field. Sophisticated momentum integral methods are used to model the boundary layer behaviour. The boundary layer models are iteratively coupled to the panel method module to predict laminar and turbulent effects. While these models are based on both analytical expressions and empirical data, they represented some of the most advanced theoretical treatments available. Figure 2.5 illustrates the results output of an XFOIL simulation. The surface pressure is illustrated by the inviscid distribution (- -) and the viscous-corrected distribution (—).

Panel methods provide remarkably accurate results rapidly, and are an attractive choice for subsonic multi-fidelity analyses [Keane and Nair, 2005]. The use of VSAERO was reported in the study of Ong *et al.* [2003] for a constrained wing design problem. Wickramasinghe *et al.* [2010] and Khurana and Winarto [2010] use XFOIL to predict airfoil performance for a wide range of operating conditions. In the optimization framework of Jones *et al.* [1998], XFOIL is used to compute the airfoil coefficients, which is subsequently passed to an aeroacoustic module to compute the sound pressure level. Despite the advantages of the panel methods, their main deficiency lies in the inability to accurately model compressible effects, which restricts their application to subsonic problems.

2.3.4 Full Potential Methods

The full potential equation describes inviscid, irrotational and compressible flow. Since the full potential equation includes all compressibility terms, solutions are valid for high



(a) Physical grid (x, y) (b) Computational grid (ξ, η)

Figure 2.6: Transformation from physical to computational domain for full potential method

Mach number flows, however the irrotational condition imposes a condition of constant entropy throughout the flowfield. This results in the isentropic flow relation across shocks and therefore does not apply to flows with strong shocks. This flaw of the full potential equation results in a general overestimation of the lift and drag. In optimization, potential flow solvers are most notably used in the transonic regime, where shocks are weak yet it is assumed that the wave drag due to shocks is more significant than the component of drag due to viscosity. Despite its inaccuracy in modelling strong shocks, full potential solvers are still robust enough to drive the search away from poor performing regions of the design space. The full potential method has been successfully used in several transonic airfoil and wing optimization frameworks [see Hicks and Henne, 1978; Vicini and Quagliarella, 1997; 1999; Obayashi *et al.*, 1997].

In this research the divergence form of the potential flow equation, using the rotated difference scheme of Jameson [1974], is solved on a two-dimensional grid for the inviscid compressible flow around supercritical airfoil sections. The grid is generated by solving a system of elliptic equations, from the work of Thompson *et al.* [1974]. The physical grid (x, y) is transformed to a finite difference computational grid (ξ, η) as shown in Figure 2.6.

2.3.5 Euler Method

For compressible flows at large Reynolds numbers, in the absence of massive flow separation, viscous and turbulence effects are limited to a thin layer close to the solid surface. When the boundary layer is sufficiently thin compared with the characteristic length of the problem, the diffusion terms can normally be neglected so that the Navier-Stokes equations are reduced to the Euler equations. Mathematically, the system of Euler equations constitutes the most complete description of inviscid flows, and can thus be used to solve all strong compressible flows (see Figure 2.7). Although the Euler equations are obviously

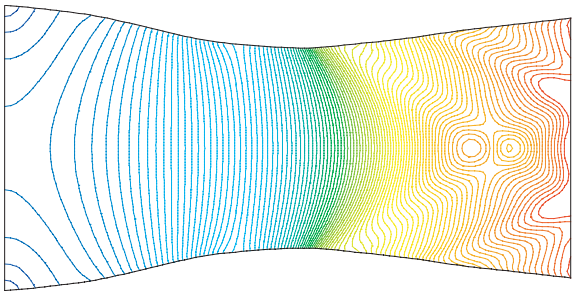


Figure 2.7: Euler simulation illustrating Mach number contours for a supersonic nozzle

not universally valid, the importance of their accurate numerical simulation resides in the dominating convective character of the Navier-Stokes equations at high Reynolds number. The two-dimensional Navier-Stokes momentum equations (2.4)–(2.5) therefore reduce to:

$$\frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho u^2)}{\partial x} + \frac{\partial(\rho uv)}{\partial y} = -\frac{\partial p}{\partial x}, \tag{2.11}$$

$$\frac{\partial(\rho v)}{\partial t} + \frac{\partial(\rho uv)}{\partial x} + \frac{\partial(\rho v^2)}{\partial y} = -\frac{\partial p}{\partial y}. \tag{2.12}$$

The Euler equations require orders of magnitude less computing time than solving the (Reynolds-averaged) Navier-Stokes equations. Euler solvers constitute the majority of modern aircraft design codes [Keane and Nair, 2005], and codes generally incorporate empirical boundary layer models to quantify the viscous effects in the linear regime.

Euler methods have been used extensively in aerodynamic design. Reuther and Jameson [1995] integrated an Euler solver into a supersonic wing-body shape optimization framework using an adjoint formulation. It was demonstrated that shock-reduced designs were identified at high supersonic Mach numbers. A comprehensive study on the use on the adjoint approach for aerodynamic shape optimization using an unstructured Euler solver is also presented [Anderson and Venkatakrishnan, 1997, and references therein]. Elliott and Peraire [1996] used an unstructured Euler solver for a number of case-studies, demonstrating the feasibility of the Euler formulation for optimization studies. Obayashi *et al.* [2000b] utilized an Euler solver for the evaluation of the supersonic performance of wings, where simulations were executed in parallel for computational time savings. In a later study, Huyse and Michael [2001] adopted an unstructured Euler solver for multi-point airfoil optimization at varying Mach number range. However, it was concluded from this study that a boundary layer correction was necessary in order to accurately predict surface pressure. Quagliarella and Vicini [2001] presented a comprehensive optimization framework of a high-lift airfoil configuration using an Euler-based solver, coupled with the compressible boundary layer equations to model the viscous effects. Most commercial CFD codes can be used to solve the compressible Euler equations. In this research, the

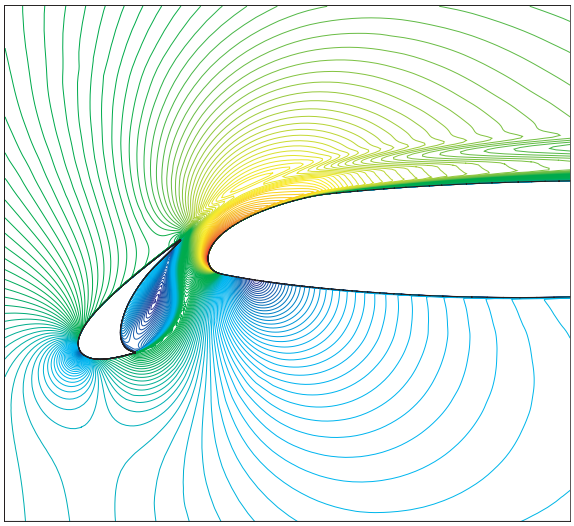


Figure 2.8: RANS simulation of a slat configuration illustrating confluent boundary layer

compressible Euler equations are solved using the commercial finite volume code ANSYS Fluent. The Euler solver is used for flows of high Mach number and Reynolds number, where viscous effects are dominated by inertial forces.

2.3.6 (Reynolds-Averaged) Navier-Stokes

The most common simplification to the full Navier-Stokes equations are the Reynolds-averaged Navier-Stokes (RANS) equations [Anderson *et al.*, 1984], which describe (un)steady, compressible and viscous flows. In order to take into account the turbulent motion of the fluid, the statistical averaging procedure (or the Reynolds decomposition) of each variable is used. The value of the generic variable g is therefore comprised of two parts such that,

$$g = \bar{g} + g' \tag{2.13}$$

where $\bar{g}$ is the time-averaged value over the given time interval, and g' refers to the fluctuating component due to turbulence. When the dependent variables in the Navier-Stokes equations (2.6)–(2.7) are replaced with their time-averaged and fluctuating components, additional apparent stress and heat transfer terms appear [Anderson *et al.*, 1984]. Consider the Navier-Stokes momentum equations (2.4)–(2.5) with Reynolds decomposition:

$$\frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho u^2)}{\partial x} + \frac{\partial(\rho uv)}{\partial y} = -\frac{\partial p}{\partial x} + \frac{1}{Re}\left[\frac{\partial \bar{\tau_{xx}}}{\partial x} + \frac{\partial \bar{\tau_{xy}}}{\partial y}\right] - \left[\frac{\partial \bar{\tau'_{xx}}}{\partial x} + \frac{\partial \bar{\tau'_{xy}}}{\partial y}\right], \tag{2.14}$$

$$\frac{\partial(\rho v)}{\partial t} + \frac{\partial(\rho uv)}{\partial x} + \frac{\partial(\rho v^2)}{\partial y} = -\frac{\partial p}{\partial y} + \frac{1}{Re}\left[\frac{\partial \bar{\tau_{xy}}}{\partial x} + \frac{\partial \bar{\tau_{yy}}}{\partial y}\right] - \left[\frac{\partial \bar{\tau'_{xy}}}{\partial x} + \frac{\partial \bar{\tau'_{yy}}}{\partial y}\right]. \tag{2.15}$$

It is observed that Eqs. (2.14)–(2.15) are similar to Eqs. (2.4)–(2.5) with the addition of the fluctuating shear stress components. To model these new apparent stress terms (commonly

known as the Reynolds stresses) requires additional equations. These additional equations or *closure* models make empirical assumptions regarding the relationship between the time-averaged variables and the apparent turbulent quantities.

To appreciate the domain of application of RANS modelling, consider the simulation with the Spalart-Allmaras turbulence closure model [Spalart and Allmaras, 1992] for the three-element airfoil configuration shown in Figure 2.8. The velocity contour demonstrates the merging of the wake from the forward element with the surface boundary layer of the aft element to generate a confluent boundary layer. To achieve this practical level of accuracy for aerodynamic design, it would be essential to solve the RANS equations. Since finer meshes are required to assure sufficient accuracy, parallel computing is of particular significance in aerodynamic design via RANS modelling.

The use of RANS for aerodynamic design optimization was popularized by the work of Jameson *et al.* [1998]. In these (and previous) papers, the authors present the solution of the adjoint problem for the RANS equations. Since the introduction of this study, the use of RANS in aerodynamic design frameworks is a recurring theme and appears in notable (among others) studies in the literature (see, for example, Elliott and Peraire [1997]; Anderson and Bonhaus [1999]; Nemec *et al.* [2004]; Kim *et al.* [2004]; Jeong *et al.* [2005b]; Epstein *et al.* [2009]; Jahangirian and Shahrokhi [2009]). While these papers focus on the steady-state approach for aerodynamic design optimization, unsteady design frameworks have also been reported [see Rumpfkeil and Zingg, 2010].

Commercial CFD codes can be used to solve the compressible RANS equations. In this research, the compressible RANS equations are solved using the commercial finite volume code ANSYS Fluent. The one-equation Spalart-Allmaras turbulence closure model is utilized [Spalart and Allmaras, 1992]. The advantage of this model over two-equation turbulence models is its robust convergence rate which makes it suitable for aerodynamic flows over complex geometries.

2.4 Summary

This chapter has introduced the aerodynamic design architecture and provided an insight into the role of numerical modelling and shape parameterization in the aerodynamic design framework. The shape parameterization techniques and numerical models used in this research have been described and the effects of these elements on the efficiency and success of the design process have been identified. It was established that the shape parameterization method essentially governs the dimensionality of the problem and the attainable

shapes, whereas the objective flow solver dictates the overall fidelity of the process. In the next chapter the most important element of the design architecture is introduced, namely the optimization algorithm.

Chapter 3

Design and Optimization

The aerodynamic design architecture is a synthesis of numerical methods for analysis and optimization techniques. In the previous chapter, a description on the numerical elements of the aerodynamic design architecture was provided. While these elements both influence the efficiency of the optimization loop, arguably the most important element is the optimizer itself. This chapter presents a discussion on the field of optimization and the characteristics which classify an optimization problem. The concepts and techniques which are especially applicable to this research are introduced, laying the foundations for the proposed methodology documented in later chapters.

3.1 Introduction to Optimization

Optimization problems may be characterized in a number of different ways. To appreciate the optimization process and the techniques developed to solve optimization problems, it is beneficial to familiarize oneself with the terminology and concepts liberally used in this thesis to characterize problems.

The aim of an optimization process in principle is to identify the optimum solution to an objective function. The objective function represents the quantity to be optimized, and it is generally up to the discretion of the designer to formulate the objective such that it reflects the design requirements. If the problem involves one objective function, then it is referred to as a single-objective optimization problem. If the problem consists of more than one objective function it is considered multi-objective (see Section 3.4), which is the focus of the current research. The objective function is optimized by systematically modifying the input design variables, such that a candidate solution refers to a specific set of inputs. The methodology for specifying variable modifications is essentially what characterizes an optimization algorithm. Since all optimization methods solve problems iteratively,

a convergence criterion is necessary[1]. While this should ideally be based on attaining the optimum solution, in design it is not uncommon that convergence is assumed once a computational budget has been exceeded (termed a *stopping* criterion). The remainder of this section is devoted to the fundamental concepts of the generic optimization problem.

3.1.1 Optimality

Let $\mathcal{S} \in \mathbb{R}^n$ denote the n-dimensional design space and let $\mathbf{x} = \{x_1, x_2, \ldots, x_n\} \in \mathcal{S}$ be a vector of design variables. Without loss of generality, it is assumed that the single objective function $f : \mathbb{R}^n \to \mathbb{R}$ is to be minimized. The definition of the global optimum $\mathbf{x}^*$ of the objective function $f(\mathbf{x})$ is therefore,

$$f(\mathbf{x}^*) < f(\mathbf{x}) \quad \forall \mathbf{x} \in \mathcal{S} \text{ and } \mathbf{x} \neq \mathbf{x}^*. \tag{3.1}$$

Optimization problems are often multi-modal and possess multiple optima. Apart from the case where all optima are equally global, there is generally only one single global optimum and multiple local optima which are characterized as either strong or weak. A solution $\mathbf{y}^*$ is a strong local minimum of $f(\mathbf{x})$ if

$$f(\mathbf{y}^*) < f(\mathbf{y}) \quad \forall \mathbf{y} \in \mathcal{N}(\mathbf{y}^*, \eta) \text{ and } \mathbf{y} \neq \mathbf{y}^*, \tag{3.2}$$

where $\mathcal{N}(\mathbf{y}^*, \eta) \subset \mathcal{S}$ is the neighbourhood of $\mathbf{y}^*$ which extends some arbitrary distance η from $\mathbf{y}^*$. For $\mathbf{y}^*$ to be a weak local minimum only the inequality of Eq. (3.2) need be satisfied, such that $f(\mathbf{y}^*) \leq f(\mathbf{y})$ under the same conditions. Figure 3.1 shows a one-dimensional representation of a multi-modal function landscape. The global optimum is the solution $f(x^*)$ and the solution $f(x')$ is a strong local optimum. The cluster of solutions represented by $f(x'')$ are weak local optima.

3.1.2 Constraints

Most design optimization problems are restricted due to constraints, which may either confine the boundaries of the design space, or exclude a solution from being considered if they are violated. Consider a simple airfoil shape optimization problem, where it is sought to minimize the drag coefficient during cruise flight. A boundary constraint may be in the form of a threshold to the minimum thickness-to-chord variable, in order to carry sufficient fuel. A conditional constraint on the other hand may dictate that the lift coefficient must be greater than or equal to a specific value, to ensure steady level flight.

[1] With the exception of classical convex critical point theory

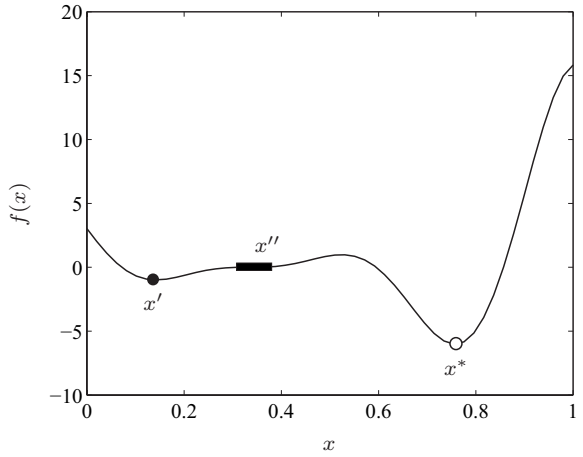


Figure 3.1: One-dimensional representation of a multi-modal objective landscape

Boundary Constraints

Boundary constraints define the domain of the n-dimensional search space. These constraints are of the form:

$$x_{\min,j} \leq x_j \leq x_{\max,j}, \qquad \forall j = 1, \ldots, n.$$

Boundary constraints restrict the domain of the objective functions within which an optimum may be located. Consider the one-dimensional objective function shown in Figure 3.2. If the boundary of x is defined such that $x \in [0, 1]$ then the solution $f(x')$ is the global optimum and $f(x^*)$ is a local optimum. However, if the boundary is now defined such that $x \in [0, 0.5]$, then the solution at $f(x^*)$ is the global optimum and $f(x')$ is now an infeasible solution. It is important to note that although the solution $f(x')$ is the global *minimum*, within the optimization constraints the solution $f(x^*)$ is the global *optimum*.

The example in Figure 3.2 illustrates the necessity to appropriately define the boundaries of design variables to facilitate the optimization process. Defining the boundary such that $x \in [0, 1]$ causes the function $f(x)$ to be multi-modal, requiring an increased search effort to converge to the global optimum. If the boundary were defined as $x \in [0, 0.5]$ the function becomes uni-modal, however the optimization process will overlook better performing areas of the design space. It would be ideal to define the boundary $x \in [0.5, 1]$, such that the problem becomes uni-modal and convex, and the global optimum is assured to be the best performing solution. Design variable domains may be restricted to provide robust designs, at the expense of limited flexibility and the risk that better performing areas are bypassed completely. On the other hand, applying no restrictions will provide greater flexibility to explore improved designs at the expense of an increased computational effort. In design it is ideal to avoid any unnecessary computational effort, and

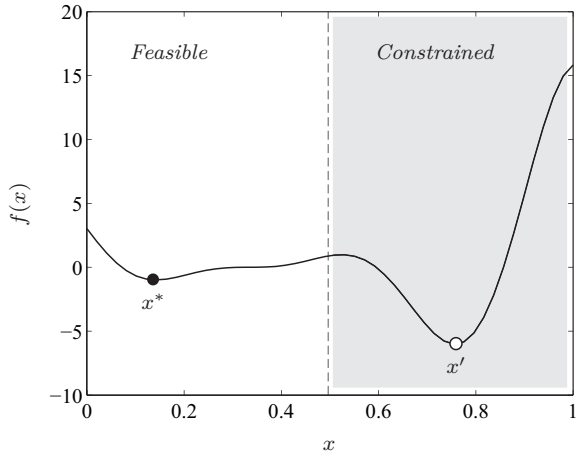


Figure 3.2: Shift of global optimum as a result of boundary constraints

thus it is beneficial to have some prior understanding of the design space, such that areas which are *likely* to provide the optimum are isolated. For example, in the subsonic airfoil design framework of Khurana and Winarto [2010], the design space domain was defined by pre-screening existing profiles which perform well for the given operating conditions.

Conditional Constraints

Conditional constraints are represented as inequality or equality functions of the design variables which exclude solutions if they are violated. They are of the form:

$$g_k(\mathbf{x}) \geq 0, \ \forall k = 1, \ldots, n_k,$$
$$h_p(\mathbf{x}) = 0, \ \forall p = 1, \ldots, n_p,$$

where n_k and n_p are the number of inequality and equality constraints respectively. Equality constraints may be treated as a pair of inequality constraints, such that $h(\mathbf{x}) = 0$ is transformed to $h(\mathbf{x}) \leq \epsilon$ and $h(\mathbf{x}) \geq -\epsilon$, where ϵ represents some threshold parameter. Consider the same one-dimensional objective function shown in Figure 3.3. It is sought to minimize $f(x)$ subject to the linear inequality constraint of the generic form $A \cdot x \leq B$. It is clearly shown that the global minimum of $f(x)$ will violate the constraint. As a result, the global optimum of the constrained optimization problem is now considered to be the solution $f(x^*)$.

Where boundary constraints are easily handled by restricting the design space domain, conditional constraints must generally be solved simultaneously with the optimization problem. Furthermore, conditional constraints generally refer to specific design requirements that must be maintained or achieved, providing the designer limited flexibility to alleviate or avoid their effect.

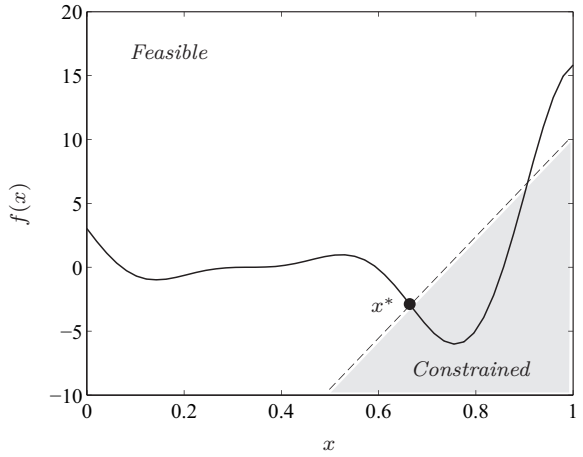


Figure 3.3: Illustration of a constraint optimization problem

3.2 Optimization Techniques

Optimization algorithms repeatedly call the objective and constraint functions to maintain the optimal search direction. Since in aerodynamic design these functions are represented by computationally expensive flow solvers (see Section 2.3) it is essential that the optimizer be as efficient as possible, to achieve convergence by expending the least computational effort. The characteristics of the optimization problem are the primary factors that influence the choice of optimization algorithm. For design applications it is also necessary to consider factors such as practicality, and whether the aim is to achieve a global design or to locally improve on an existing design. This section describes and evaluates some of the more popular optimization techniques applied for aerodynamic design applications.

3.2.1 Gradient Methods (GM)

The Gradient Methods (GM) exploit the objective function gradient to establish the optimal search direction. A single candidate solution is iteratively updated by modifying the direction and step length of the design variables. GM techniques are influenced by the dimensionality of the design space, since for each design variable the gradient must be approximated in order to define the optimal direction. The computational viability of these methods is thus largely dependent on the cost of evaluating the objective, and the method used to approximate the function gradient [Keane and Nair, 2005]. GM techniques almost exclusively rely on the condition that the objective function and constraints are differentiable and convex, which suggests that near a minimum the objective function is approximated as an ellipsoid[2]. An example of a convex function is illustrated in Figure 3.4.

[2]This is a characteristic of the minimum of any smooth and differential function (if the region near a minimum is represented as a Taylor series expansion) [Keane and Nair, 2005]

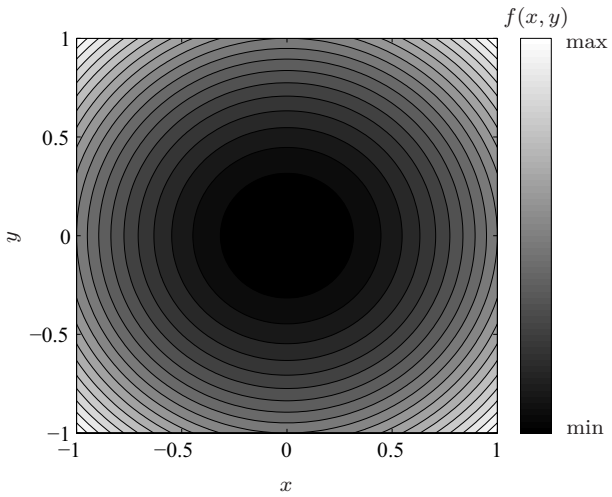


Figure 3.4: *De Jong's* convex function in two-dimensions

GM are deterministic optimizers, since they always converge to the same solution based on a given starting point. The exploitation characteristics of GM allow them to quickly converge to the mathematically exact optimum (neglecting round-off error) but only within the local neighbourhood of the starting point location. Hence if the search is initialized from an ill-defined starting point, the search will converge to an inferior minimum[3]. The use of GM is therefore highly problem dependent. If the problem is continuous, differentiable, unimodal and convex then GM is ideal due to the accelerated convergence properties. However, in most engineering design applications this is generally not the case since the objective landscape is rarely unimodal, and is restricted by constraint boundaries or discontinuities in the function landscape [Keane and Nair, 2005].

Keane and Nair [2005] provide a comprehensive review of GM techniques, and the different interpretations of the gradient information to determine the optimal search direction and step-length. GMs are proven and effective techniques for optimization, however aerodynamic design problems are generally categorized by a large number of design variables and therefore their application may be impractical due to the expensive gradient vector evaluation. It was not until the introduction of the adjoint methods by Jameson [1988] that they became popular in the aerodynamic design literature. An adjoint method refers to the use of the flow equations to compute the objective function gradient. Since the adjoint approach is independent of the number of design variables, the computational burden of optimizing multi-dimensional problems is significantly alleviated.

Giles and Pierce [2000] and Jameson [2003] present reviews of the development and application of the adjoint approach for aerodynamic design. Adjoint-based gradient op-

[3]It can be argued that multiple starting point strategies may mitigate this issue

timization is now a common theme in aerodynamic shape design frameworks, notably in the work of Jameson *et al.* [1995; 1998; 2004], Anderson and Venkatakrishnan [1997], and Zingg *et al.* [2002; 2004; 2006; 2010]. The introduction of the adjoint approach allows GM to be a viable and practical technique for aerodynamic design. If the aim of the optimization process is to locally improve on an existing design, then evidently adjoint-based gradient optimization is ideal. However for global optimization of multi-modal landscapes the use of any gradient-based approach is ultimately questionable, since premature convergence to an inferior optimum is highly probable. Furthermore, since only one candidate is progressed through the optimization loop, their domain of application is limited. For example, with reference to this research in particular, GM cannot be used to identify multiple trade-off solutions to a multi-objective problem. The stochastic population-based algorithms address the limitations of GM and are the preferred technique for this research.

3.2.2 Evolutionary Algorithms (EAs)

Evolutionary Algorithms (EAs) are stochastic methods inspired by evolutionary theory. EAs maintain a population of candidate solutions, which either compete or collaborate to identify optimum solutions over a number of iterations (generations). This considerably broadens their domain of application, facilitating also the explicit treatment of multiple objectives (see Section 3.5.2). EAs are global search techniques, since they simultaneously explore several promising areas of the design space. The evolutionary operators are stochastic in nature and thus the probability of premature convergence to inferior optima is significantly alleviated. In comparison to local optimizers, EAs are exploration-based techniques. Hence they are proficient in identifying promising areas of the design space, yet this is generally at the expense of slow convergence rates and poor precision in the exploitation of individual optima [Grosan *et al.*, 2007]. This is the primary uncertainty in applying EAs to aerodynamic design, since computational complexity is a prohibiting factor, and the use of parallel computing becomes a necessity [Keane and Nair, 2005].

EA techniques do not rely on the gradient vector to determine the search direction, and do not require that the function is continuous or differentiable. EAs are therefore applicable to all problem classifications, and are ideal for multi-modal and discontinuous functions. An example of a multi-modal function is illustrated in Figure 3.5. EAs cannot absolutely guarantee to converge to the global optimum, yet they can explore several basins of attraction simultaneously. A typical search process involves the random initialization of a population of individuals, eliminating the requirement to provide a starting point. The objective and constraint functions are evaluated for each individual, in the form of a *fitness* function. Evolutionary operators are subsequently used to define a new

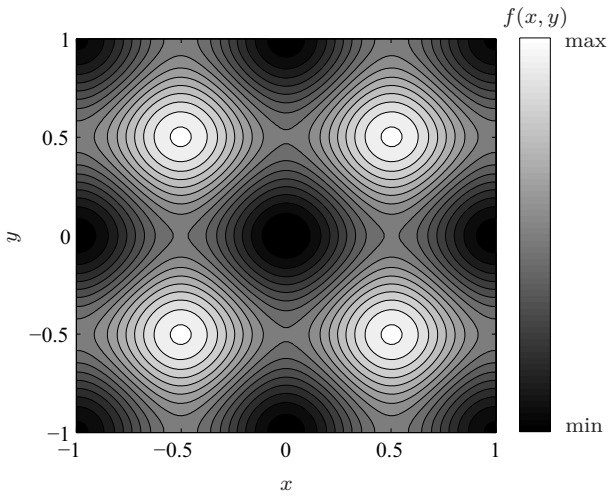
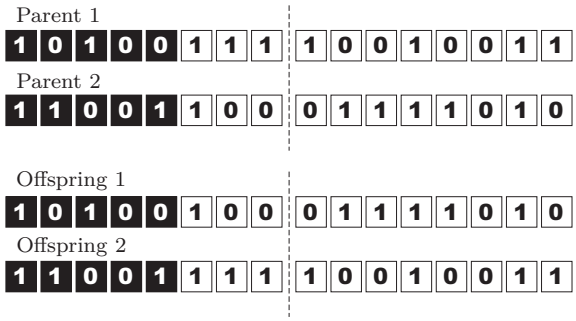


Figure 3.5: *Rastrigin's* multi-modal function in two-dimensions

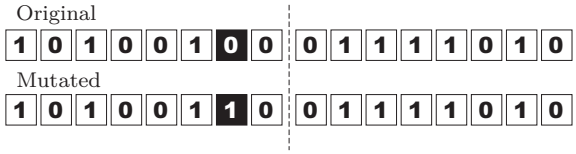
population of individuals, which inherent the successful attributes of the previous generation and reject poorer solutions. These operators introduce search diversity whilst attempting to balance exploitation of promising areas of the design space. The algorithms which fall under the class of EAs differ in their origins and their interpretation of *evolving* a population over time. For example, simulated annealing [Kirkpatric *et al.*, 1983] originates from the process of annealing in metallurgy. Differential evolution [Storn and Price, 1997] or genetic algorithms [Goldberg, 1989] mimic genetic evolution and Darwinian *survival of the fittest*. Other algorithms originate from the field of computational swarm intelligence [Engelbrecht, 2005], where a particle swarm or a colony of ants collaborate to identify promising areas of the design space. Each of these algorithms have their identified merits and weaknesses[4], and may be suited to specific forms of optimization.

Aerodynamic designers and researchers are now realizing the benefits of incorporating EAs within an optimization framework, mainly for the enhanced probability to navigate multi-modal and discontinuous landscapes. For example, in the shape optimization framework of Rogalsky *et al.* [2000], differential evolution is shown to outperform more traditional optimization techniques. Wang and Damodaran [2001] demonstrate the ability of a simulated annealing algorithm to identify optimum solutions for the shape optimization of nozzles and diffusers. Their parallel computing architecture led to a significant reduction in computing time. However, by far the most popular EAs adopted for aerodynamic design in the literature is the Genetic Algorithm (GA) [see Goldberg, 1989]. GAs adopt the Darwinian theory of *survival of the fittest*, where successful individuals have a high probability to participate in reproduction for the next generation. Following this methodology, poorly

[4]Refer to Keane and Nair [2005] for a comprehensive discussion on popular evolutionary heuristics.



(a) Single-point crossover



(b) Single-bit mutation

Figure 3.6: Illustration of genetic operators

performing solutions are eliminated from the population, and fitter individuals reproduce to ensure successful offspring. In the canonical GA candidates are expressed as binary encoded strings (genomes), and genetic operators such as selection, crossover and mutation are used to evolve the population over a number of generations. For example, consider two successful solutions (parents) of two variables, where each variable is encoded by an 8-bit binary string. Figure 3.6(a) illustrates the single-point crossover operator, where a point in the genome is randomly defined where the bits are exchanged. The highlighted region indicates the bits which are exchanged between parents to produce the offspring. Figure 3.6(b) demonstrates the single-bit mutation operator, where a bit is randomly selected and exchanged. Mutation is effective to maintain diversity in the search and can alleviate the probability of premature convergence.

In the studies by Quagliarella and Della Cioppa [1994] and Yamamoto and Inoue [1995], GAs are applied for the shape optimization of transonic airfoils. Results highlight the ability of GAs to converge towards shockless airfoils, demonstrating the viability of GA for airfoil shape optimization studies. In the shape optimization framework of Holst and Pulliam [2001], GA is applied for quasi-one-dimensional nozzle design and transonic wing design. The case-studies demonstrate the robustness of GA and their insensitivity to design space noise. The approach by Quagliarella and Della Cioppa [1994] was later extended by Vicini and Quagliarella [1999] to incorporate a hybrid optimization strategy with a gradient-based optimizer. The ability of GA to explore the design space was sup-

ported by the exploitation proficiency of the GM to identify optimal solutions at a reduced computational cost. In the transonic wing shape optimization of Oyama *et al.* [2001], a real-coded adaptive range GA was implemented. The real-coded GA is an intuitive selection for design problems, and the adaptive range technique facilitates the navigation of large design spaces. These and other initiatives at exploring the viability of GA have influenced many researchers to capitalize on the advantages of evolutionary computing, establishing EA as a robust and powerful tool for aerodynamic shape optimization.

There are no clear guidelines on the use and development of EA, and where one algorithm may perform favourably for a particular problem, it may conversely fail for another. This research focuses on the development and application of a population-based heuristic known as particle swarm optimization. This paradigm (as discussed in the next section) is derived from the field of swarm intelligence, where the choreography of bird flock is emulated to identify promising areas of the design space.

3.3 Particle Swarm Optimization (PSO)

While most population-based algorithms mimic the competitive theory of survival of the fittest, the concept of maintaining a population of *collaborating* individuals is increasingly becoming popular.

The formation of hierarchies within groups of animals is a naturally occurring phenomenon and is simple to comprehend. Even humans have the intuitive tendency to appoint leaders (e.g. political leaders, military generals, etc.). Another interesting phenomenon which is more difficult to perceive, is the self-organized behaviour of groups where a leader *cannot* be identified. This is known as swarming and is evident from the flocking behaviour of birds or fish moving in unison. Studies have demonstrated [Kennedy *et al.*, 2001; Engelbrecht, 2005] that this emergent behaviour is highly coordinated and arises from the interaction of simple rules (such as neighbour proximity). The recently developed field of swarm intelligence focuses on the artificial simulation of swarming behaviour to model a wide range of applications, including optimization [Kennedy *et al.*, 2001]. Particle Swarm Optimization (PSO) is a stochastic population-based technique developed by Kennedy and Eberhart [1995] with the principles of swarm intelligence. The PSO architecture was derived from the fields of social psychology and engineering optimization. As eloquently stated by Kennedy and Eberhart in their original paper:

> *"...Why is social behaviour so ubiquitous in the animal kingdom? Because it optimizes. What is a good way to solve engineering optimization problems? Modelling social behaviour."*

The dynamics of the swarm are modelled on the social-psychological tendency of individuals to learn from previous experience and emulate the success of others. Similar to most evolutionary optimization techniques (such as GA) the swarm is initialized with a population of random individuals (particles) sampled over the design space. The particles navigate the multi-dimensional design space over a number of iterations or time-steps. Each particle maintains knowledge of its current position in the design space. This is analogous to the *fitness* concept of GA. Each particle also records its personal best position, which is where the particle has experienced the greatest success. Aside from recording personal information, each particle in the swarm also tracks the most successful position of its surrounding neighbours, known as the neighbourhood leader. Utilizing this information, each particle *collaborates* rather than competes with other particles to adjust its position in the design space by *accelerating* towards successful areas of the design space. The swarm generally converges to the most successful position as identified by any particle, yet similar to GA it is not guaranteed that this position is the global optimum.

PSO originated as a method to optimally train neural networks and it has steadily gained popularity as a global optimization technique [Kennedy *et al.*, 2001]. There are primary distinctions between PSO and GAs which have contributed towards its growing regard [Angeline, 1998; Engelbrecht, 2005]. GAs rely on three operator mechanisms to evolve the population which are selection, crossover and mutation. Each of these operators must also be configured to suit the intended application. PSO however only relies on one operator which is the particle velocity update. There is no notion of selection therefore particles do not compete with each other and they do not reproduce nor are they rejected. The absence of selection is compensated by the use of leaders to guide the swarm. In this way, a solution which initially performs poorly may possibly be on the future road to success. Despite its intricate origins, PSO with respect to GAs is also fairly simple and straightforward to implement [Reyez-Sierra and Coello Coello, 2006]. For this reason, PSO has been established as a very effective optimization tool for a wide variety of applications due to its efficient and accurate convergence rates [Trelea, 2003], and has also prompted some interest from researchers in aerodynamic design.

Perhaps one of the first recorded attempts at utilizing swarm intelligence to solve constrained aerodynamic design problems was by Ray and Liew [2002]. Swarm dynamics were combined with evolutionary selection mechanisms to maintain diversity in the search. The swarm algorithm was quite capable at handling a variety of different problems, including single and multi-objective airfoil shape optimization studies [Ray and Tsai, 2004]. In the study by Sobieszczanski-Sobieski and Venter [2004], PSO was implemented for the bi-level multi-disciplinary optimization of an aircraft transport wing. The authors concede that

while PSO may be more expensive in comparison to gradient-based techniques, having the ability to efficiently navigate the noisy aerodynamic and structural landscapes makes PSO a viable alternative. Praveen and Duvigneau [2009] report the use of PSO and local metamodels for an aerodynamic shape design case-study. A PSO-specific screening criterion was proposed, based on the expected improvement in the local memory of individual particles. Khurana and Winarto [2010] developed and implemented a novel PSO algorithm for an intricate airfoil shape design framework. The superiority of the swarm algorithm over other techniques was highlighted, thus establishing it as a feasible approach for the constrained airfoil design case-studies.

The popularity of PSO has led to increasing open source support[5]. The focus on PSO for this research is primarily motivated by the studies of Khurana and Winarto, and the most attractive feature of PSO in the opinion of the author is its elegant simplicity. With only slight variations to the algorithm it can be applied to essentially any optimization problem. Furthermore the coding of the PSO algorithm is straightforward. The remainder of this section is devoted to a discussion on the swarm topology and dynamics of the developed algorithm.

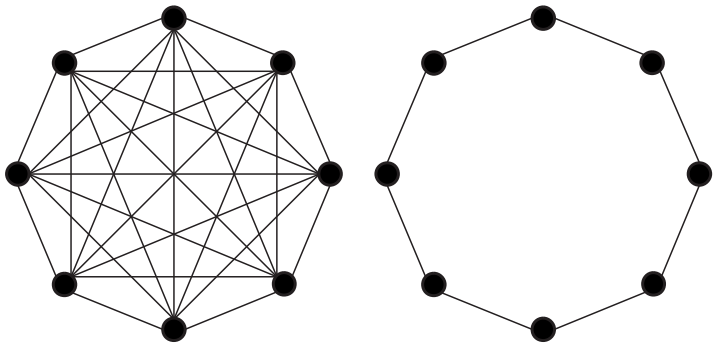
3.3.1 Swarm Topology

The change in position of a particle is dictated by its velocity. At any time-step, the velocity vector of a particle is dependent on its own personal experience and the influence of other particles. There have been numerous modifications and improvements to the canonical PSO algorithm, which affect certain search characteristics [Engelbrecht, 2005]. Of significant importance is the level of social interaction among particles, or the swarm *topology*. This is also a characteristic of naturally occurring animal behaviour, where an individual is most likely to be influenced by more successful neighbours rather than the less successful. The topology essentially characterizes the dynamics of the swarm, and is a subject of major interest among researchers [Engelbrecht, 2005]. The main variations are:

- ***gbest*:** The neighourhood of the particle is the entire swarm. Each particle can communicate with every other particle to navigate the design space. In this case each particle is attracted towards the best solution found by the entire swarm.

- ***lbest*:** Specific neighbourhoods are defined for each particle. Social interaction is confined within the local neighbourhood. Each particle is therefore attracted towards the best solution within the neighbourhood.

[5] See, for example: `http://swarmintelligence.org` and `http://particleswarm.info`



(a) Global star topology (b) Local ring topology

Figure 3.7: Illustration of main variations in swarm topology and social hierarchy

In the *gbest* topology all particles are fully-connected, providing the advantage of quickly exchanging information through the swarm. Figure 3.7(a) illustrates the *gbest* topology represented by a star structure. This structure guarantees a faster convergence rate yet consequently is more susceptible to premature convergence to local optima [Engelbrecht, 2005]. The advantage of the local topology lies in its slower convergence rate, alleviating the probability of premature convergence to an inferior optimum [Kennedy and Mendes, 2002]. Individual particles have limited global communication and the concept of a single global leader is replaced by a subset of one-to-many neighborhood leaders to encourage diversity. Figure 3.7(b) demonstrates the ring structure, where each particle communicates with only two of its immediate neighbours.

The *lbest* topology or the notion of maintaining multiple leaders within the swarm is now considered the standard, as recommended by Bratton and Kennedy [2007]. Given the problem circumstances, Bratton and Kennedy also note that even the *gbest* topology can deliver comparable performance to the *lbest* topology for complex multi-modal problems. In a multi-objective environment, the concept of the single swarm leader of the *gbest* topology is naturally replaced by a number of equally optimal trade-off solutions. This concept is comparable to the *lbest* topology, where defining multiple leaders to guide the swarm inherently incorporates more diversity in the search.

3.3.2 Swarm Initialization

A swarm of N particles is required to navigate the design space $\mathcal{S}$ bounded by $\mathbf{x}_{\min}$ and $\mathbf{x}_{\max}$. To safeguard against magnitude and scaling issues, all variables are normalized into the unit cube, such that $\mathcal{S} = [0, 1]^n$. The i-th particle in the swarm is characterized by the n-dimensional vectors $\mathbf{x}_i$ and $\mathbf{v}_i$, which are the particle position and velocity respectively.

The i-th particle is randomly initialized at time $t = 0$ by the expression

$$\mathbf{x}_i(0) = \mathbf{x}_{\min} + \mathbf{U}[0,1] \otimes (\mathbf{x}_{\max} - \mathbf{x}_{\min}), \tag{3.3}$$

where the function $\mathbf{U}[0,1]$ returns a vector of length n of uniformly distributed pseudorandom numbers $U(0,1)$. The personal best position of the i-th particle $\mathbf{p}_i$ is initialized to the position $\mathbf{x}_i$ at $t = 0$. The objective and constraint functions are calculated and fitness is assigned. While it is argued that the initial particle velocity should be zero (since in reality a swarm starts from rest), it is possible to initialize the velocity such that the particle has an initial momentum and trajectory [Engelbrecht, 2005]. Following Wickramasinghe and Li [2008], the velocity is initialized to a random value, such that

$$\mathbf{v}_i(0) = (\mathbf{x}_{\min} - \mathbf{x}_{\max}) + \mathbf{U}[0,1] \otimes (2 \cdot \mathbf{x}_{\max}). \tag{3.4}$$

It is important to note that initializing the velocity by Eq. (3.4) randomly assigns both the magnitude and direction of each vector component.

3.3.3 Particle Update

The update equations of PSO adjust the position of the i-th particle from time t to $t+1$. Significant insight into formulating a theoretical basis for the internal dynamics of PSO were pioneered in the work of Clerc and Kennedy [2002]. In their studies, particle behaviour was analyzed from an eigenvalue analysis of swarm dynamics. The relative influence of constricting exploration and exploitation characteristics of the swarm on its overall search performance was highlighted. Their constriction *type* frameworks have since formed a departure point for further studies in refining the PSO algorithm. In the constriction *type 1* framework[6], the velocity update of the i-th particle is a function of acceleration components to both the personal best position, $\mathbf{p}_i$ and the global (or neighbourhood) best position, $\mathbf{p}_g$. The updated velocity vector is given by the expression,

$$\mathbf{v}_i(t+1) = \chi[\mathbf{v}_i(t) + \mathbf{R}_1[0,\varphi_1] \otimes (\mathbf{p}_i(t) - \mathbf{x}_i(t)) + \mathbf{R}_2[0,\varphi_2] \otimes (\mathbf{p}_g(t) - \mathbf{x}_i(t))]. \tag{3.5}$$

The velocity update of Eq. (3.5) is quite complex and is composed of many quantities which affect certain search characteristics. The previous velocity $\mathbf{v}_i(t)$ serves as a memory of the previous flight direction and prevents the particle from drastically changing direction and is referred to as the inertia component. The cognitive component of the update equation $(\mathbf{p}_i(t) - \mathbf{x}_i(t))$ quantifies the performance of the i-th particle relative to past performances. The effect of this term is that particles are drawn back to their own best

[6]The constriction *type 1* framework of Clerc and Kennedy is the recognized standard [see Bratton and Kennedy, 2007]

positions, which resembles the tendency of individuals to return to situations where they experienced most success. The social component $(\mathbf{p}_g(t) - \mathbf{x}_i(t))$ quantifies the performance of the i-th particle relative to the global (or neighbourhood) best position. This resembles the tendency of individuals to emulate the success of others.

The two functions $\mathbf{R}_1[0, \varphi_1]$ and $\mathbf{R}_2[0, \varphi_2]$ return a vector of uniform random numbers in the range $[0, \varphi_1]$ and $[0, \varphi_2]$ respectively. The constants φ_1 and φ_2 are equal to $\varphi/2$ where $\varphi = 4.1$. This randomly affects the magnitude of both the social and cognitive component. Once the particle velocity is calculated, the particle is displaced by adding the velocity vector (over the unit time step) to the current position,

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1). \tag{3.6}$$

The balance between exploration and exploitation is a critical feature of global optimization techniques. In PSO, this trade-off is heavily dependent on the magnitude of velocity. The velocity magnitude should initially be larger so unexplored areas are quickly sampled. As the algorithm identifies a promising area it is ideal to reduce the magnitude of velocity so that the area is exploited and particles are less susceptible to diverge from the search path. In PSO this balance is managed by dynamically scaling the inertia component or clamping the velocity magnitude [Engelbrecht, 2005]. In the constriction *type 1* framework of Clerc and Kennedy [2002], the velocity vector is constricted by the constant,

$$\chi = 2/|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|. \tag{3.7}$$

This factor applies a dampening effect as to how far the particle explores within the search space and is a dynamic way to ensure a stable convergence rate [Engelbrecht, 2005]. While use of the constriction coefficient eliminates the need to clamp the velocity, it was empirically demonstrated that if velocity clamping and constriction are used together, more efficient convergence rates are possible [Eberhart and Shi, 2000]. The j-th velocity component of the i-th particle is clamped as shown in Algorithm 1.

3.3.4 Boundary Conditions

Particle flight should ideally be confined to the feasible design space. However it may occur during flight that a particle involuntary violates the boundaries of the design space. There are a variety of methods for handling boundary constraints with PSO [Engelbrecht, 2005]. While it is suggested that particles which leave the confines of the design space should simply be ignored [Bratton and Kennedy, 2007], it is possible to restrict the movement of particles such that they remain within the feasible design space without affecting the flight trajectory. The *bounce-back* method considers the calculated flight trajectory and adjusts

Algorithm 1 Velocity clamping

```
for each particle x_i do
  for each variable j do
    if v_{i,j} > x_{max,j} − x_{min,j} then
      v_{i,j} = x_{max,j} − x_{min,j}
    else if v_{i,j} < x_{min,j} − x_{max,j} then
      v_{i,j} = x_{min,j} − x_{max,j}
    end if
  end for
end for
```

Algorithm 2 Boundary violation

```
for each particle x_i do
  for each variable j do
    if x_{i,j}(t+1) > x_{max,j} then
      x_{i,j}(t+1) = x_{i,j}(t) + rand() × (x_{max,j} − x_{i,j}(t))
    else if x_{i,j}(t+1) < x_{min,j} then
      x_{i,j}(t+1) = x_{i,j}(t) + rand() × (x_{min,j} − x_{i,j}(t))
    end if
  end for
end for
```

the variables which violate the boundaries. The process is demonstrated in Algorithm 2. It is important to note that the boundary condition is applied to the j-th dimension. If the j-th dimension is violated then only the j-th variable is corrected. This is particularly useful for many-dimensional problems, where penalizing all variables as a result of a boundary violation could potentially disrupt the search path of the particle.

3.4 Multi-objective Optimization

Aerodynamic design problems are often characterized by several interacting or conflicting requirements, which must be satisfied simultaneously. For example, consider the optimal design of a high-lift system for a transport aircraft wing. In this case, the aerodynamic requirement is simply to achieve maximum lift. In reality, the high-lift system will also have a strong impact on the wing weight, as well as the intricacy of system actuation. There is an important trade-off between these requirements: to maximize flight performance, a complex high-lift system is required. However, increasing the complexity of the system compromises the structural integrity, resulting in an increase in weight. Trade-offs between different requirements is a characteristic of Multi-Objective Problems (MOP). The current

research focuses exclusively on the development of a design framework for MOP. This section introduces the fundamental concepts of Multi-Objective Optimization (MOO), and reviews the techniques for solving multi-objective problems in aerodynamic design.

3.4.1 Problem Definition

Let $\mathcal{S} \in \mathbb{R}^n$ denote the n-dimensional design space and and let $\mathbf{x} = \{x_1, x_2, \ldots, x_n\} \in \mathcal{S}$ denote the decision vector with lower and upper bounds $\mathbf{x}_{\min}$ and $\mathbf{x}_{\max}$ respectively. Let the objective function $f_i(\mathbf{x})$ be defined as $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ be the i-th component of the objective vector $\mathbf{f}(\mathbf{x}) = \{f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})\}$ where m is the number of objectives. The generic multi-objective minimization problem is thus defined as:

$$\min \ \mathbf{f}(\mathbf{x}) \ \text{ subject to: } c_k(\mathbf{x}) \geq 0, \ \forall k = 1, \ldots, p. \tag{3.8}$$

Where p is the number of inequality constraints c_k. Equality constraints may be appropriately represented as two inequality constraints, as discussed in Section 3.1.2.

3.4.2 Pareto Optimality

For a single-objective problem, the global optimum is the solution which satisfies the condition $\min f_i(\mathbf{x})$. For a MOP the definition of the optimum must be redefined since in the presence of conflicting objectives, improvement in one objective may cause a deterioration in another. Typically there is no single global solution and it is often necessary to identify a set of trade-off solutions, which can all be considered equally optimal. Finding a set of these solutions, and quantifying how much better these solutions are compared to others, is the aim when formulating and solving a MOP. A solution is termed globally non-dominated or Pareto optimal[7] if the value of any objective cannot be improved without deteriorating at least one other objective. Let the candidate solutions, $\mathbf{a}$ and $\mathbf{b} \in \mathcal{S}$. The candidate $\mathbf{a}$ *dominates* the candidate $\mathbf{b}$ (denoted by $\mathbf{a} \prec \mathbf{b}$) if,

$$\forall i = 1, \ldots, m \ \ f_i(\mathbf{a}) \leq f_i(\mathbf{b}) \ \wedge \ \exists i : \ f_i(\mathbf{a}) < f_i(\mathbf{b}). \tag{3.9}$$

The concept of dominance is illustrated in Figure 3.8 for a bi-objective minimization problem. The shaded area denotes the area of objective vectors dominated by $\mathbf{a}$. A decision vector $\mathbf{a}^*$ is therefore Pareto optimal if there is no other feasible decision vector $\mathbf{a} \neq \mathbf{a}^* \in \mathcal{S}$ such that $\mathbf{f}(\mathbf{a}) \prec \mathbf{f}(\mathbf{a}^*)$. A locally non-dominated solution within a neighbourhood can similarly be defined as discussed in Section 3.1.1.

[7] The term is named after the 19th century Italian economist Vilfredo Pareto, who used the trade-off concept in income distribution [Pareto, 1927]

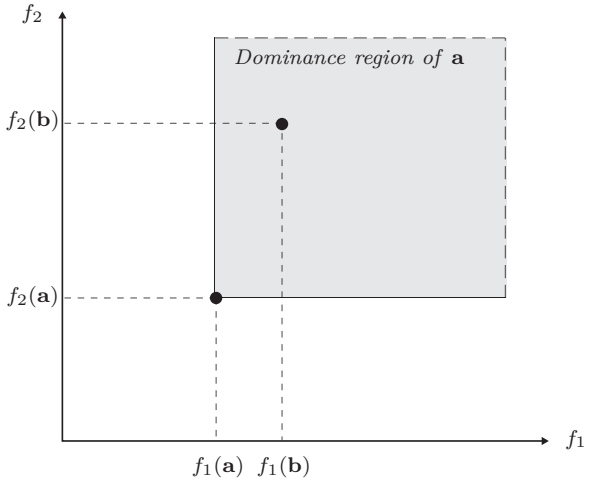
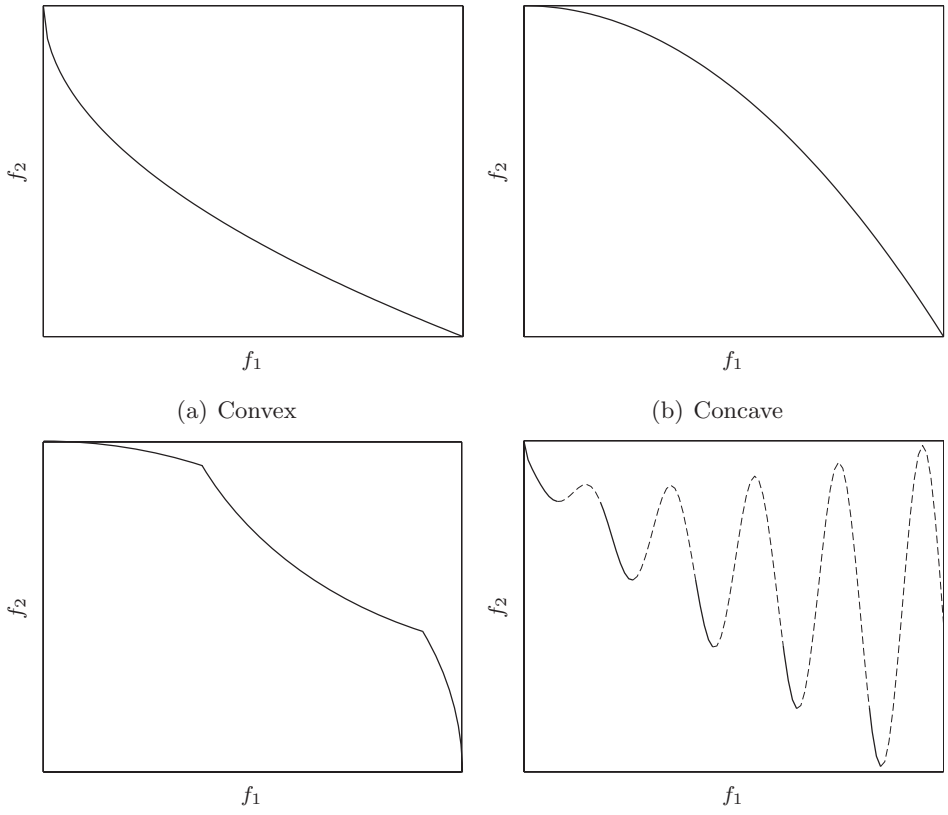


Figure 3.8: Illustration of dominance on a bi-objective landscape

3.4.3 Pareto Front

The Pareto front contains the objective vectors which correspond to all Pareto-optimal solutions. Identifying the true or exact Pareto front is generally computationally prohibitive [Engelbrecht, 2005]. The aim when solving a MOP is therefore reduced to approximating the full or partial Pareto optimal front, such that the distance to the Pareto front is minimized, and the Pareto optimal set is as diverse as possible. A key characteristic of multi-objective methods is the nature of the solutions that they provide [Arora, 2004]. Some methods may always yield Pareto optimal solutions but may omit certain solutions along the Pareto front. Alternatively, other methods may capture all possible solutions on the Pareto front but may also provide false Pareto-optimal solutions [Arora, 2004]. The tendency of a particular method to result in false Pareto optima, or being incapable of identifying all solutions is not only influenced by the method itself, but also by the problem. For example, the Pareto landscape is of particular importance in MOO, and in some cases restricts the attainability of Pareto optimal solutions [Deb, 2001; Marler and Arora, 2004]. Figure 3.9 illustrates common Pareto landscapes for bi-objective problems.

There is only one Pareto front to a given problem, which (following the optimization terminology) could be considered the *global* non-dominated front. The concept of the global non-dominated front therefore suggests the existence of false or *local* fronts [see Deb, 2001]. For single-objective multi-modal problems, algorithms are developed to identify the global optimum and avoid premature convergence to local solutions. Similarly, multi-objective algorithms aim to identify the closest approximation to the true (i.e. global) Pareto front, and avoid premature convergence to local inferior fronts.



(a) Convex (b) Concave

(c) Convex and Concave (d) Discontinuous

Figure 3.9: Illustration of example Pareto landscapes for bi-objective functions

3.5 Solving Multi-objective Problems

From a design perspective, the primary aim of MOO is to obtain Pareto optimal solutions which are in the preferred interests of the designer, or best suit the intended application. Methods for solving MOP are therefore characterized by how the designer preferences are articulated [Marler and Arora, 2004]. There are three generic classes of methods for solving MOP [Fonseca and Fleming, 1998]:

- *A priori* methods: The preferences of the designer are expressed prior to the optimization process, generally resulting in one identified trade-off.

- *A posteriori* methods: The algorithm first identifies a set of non-dominated solutions, subsequently providing the designer greater flexibility in selecting the most appropriate solution.

- *Interactive* methods: The decision making and optimization processes occur at interleaved steps, and the preferences of the designer are interactively refined.

The *a priori* strategy requires the designer indicate the relative importance of each objective before performing the optimization. Notable methods are the weighted aggregation methods, the Lexicographic method [Marler and Arora, 2004], physical programming [Messac, 1996], as well as the goal programming approach, which makes specific reference to a target solution [see, for example Sobieszczanski-Sobieski and Haftka, 1996, and references therein]. The *a priori* methods simplify the MOP at the expense of limited flexibility, since multiple runs are required to generate a set of interesting trade-offs. The *a posteriori* methods provide maximum flexibility to the designer, at the expense of greater computational effort. Generally these methods involve explicitly solving each objective using the dominance criteria, a concept which is ideal for population-based EA [Fonseca and Fleming, 1995; Deb, 2001; Coello Coello, 2007]. However, there have been some notable exceptions in the literature, such as the normal constrained method of Messac *et al.* [2003]. The *interactive* methods involve the progressive articulation of preferences, which originates from the multi-criteria decision making literature [Ehrgott and Gandibleux, 2002]. The optimization and decision making processes are interleaved, exploiting the intermediate information provided by the optimizer to refine preferences [Fonseca and Fleming, 1998].

This section presents a discussion on the multi-objective methods adopted within the scope of aerodynamic design, namely the weighted sum approach (which is generally solved via gradient-based methods) and evolutionary multi-objective optimization. A comprehensive introduction to the use of PSO for MOO is subsequently presented. The concepts described hereafter are fundamental to introducing the field of preference-based optimization, which constitutes a significant component of this research.

3.5.1 Weighted Aggregation

A simple and common approach to multi-objective aerodynamic design is to adopt the weighted sum approach, where all objectives are aggregated into a single scalar,

$$F = \sum_{i=1}^{m} \omega_i f_i(\mathbf{x}). \tag{3.10}$$

Where ω is a vector of weights typically set by the designer such that $\sum_{i=1}^{m} \omega_i = 1$. If all weight values are positive, then a solution to the weighted sum approach will likely be Pareto optimal [Marler and Arora, 2004]. The relative weights are effectively a method of stipulating designer preferences *a priori*, since the weights reflect the relative priority of each objective. For example, consider a case-study with two conflicting objectives. The simplest strategy in assigning weights would be the expression,

$$\min_{\mathbf{x}} \; F = \omega \cdot f_1(\mathbf{x}) + (1 - \omega) \cdot f_2(\mathbf{x}) \tag{3.11}$$

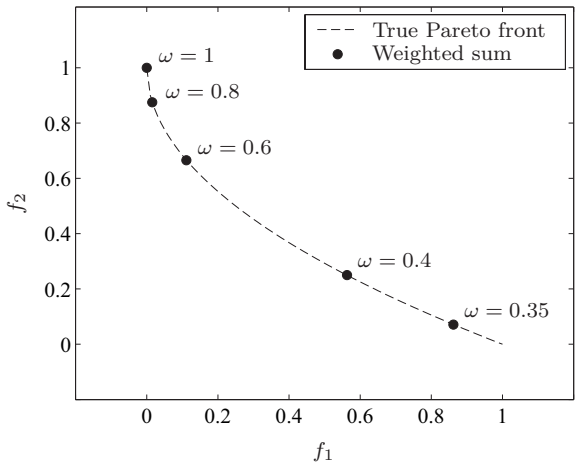


Figure 3.10: Illustration of the weighted sum approach for a convex Pareto front

Eq. (3.11) reduces to the single-objective optimization problem $\min f_1(\mathbf{x})$ as the weight value $\omega \to 1$. Any value of $\omega \in (0, 1)$ will therefore result in a different compromise between both objective functions[8]. Figure 3.10 illustrates the weighted sum approach for a bi-objective optimization problem. The trade-offs on the convex Pareto landscape are obtained by incrementing the value of ω in Eq. (3.11). It is observed that linear increments in ω do not correspond to a uniform spread of solutions.

In the aerodynamic design literature, Drela [1998] introduced the weighted sum approach for multi-objective airfoil shape optimization. Drela observed that the real deficiency of traditional airfoil design philosophies for a single operating condition was the often significant performance degradation at off-design conditions. To address this inadequacy, the concept of multi-point optimization was introduced where a range of operating conditions are considered, such that optimal performance is achieved throughout the desired range. The relative importance of each operating condition is established through weights, however their appropriate values are not intuitive and cannot be easily estimated without prior experience. The study by Drela laid the foundations for utilizing the weighted sum approach for MOO. Correlations between the geometrical degrees of freedom and the number of operating conditions were established, and the potential limitations of the approach were identified. Since then, researchers have addressed the topic of multi-point optimization in various contexts within aerodynamic design.

Li *et al.* [2002] provides a comprehensive discussion on robust optimization using the multi-point strategy, and presents a case study on airfoil wave drag optimization. In the paper by Nemec *et al.* [2004] a Newton-Krylov GM is applied for multi-objective

[8]It is interesting to note that the weighted sum approach can identify trade-off solutions to a multi-objective problem despite no explicit reference to the concept of dominance.

aerodynamic shape optimization using the weighted sum approach. Through their implementation of the discrete adjoint method, the computational expense in calculating the gradient is significantly reduced, providing an effective approach for aerodynamic design problems with multiple objectives. Furthermore, the proposed algorithm was used to compute a Pareto front based on competing objectives, where the results were validated using a dominance-based EA (see Section 3.5.2). The weighted-sum strategy using the gradient-based method was demonstrated to be convincingly more efficient than the evolutionary algorithm for the particular case-study.

The concept of multi-point aerodynamic optimization was further developed by the work of Zingg and Elias [2006]. In their study a technique was proposed for automatically choosing sampling points within the operating range and their weights to obtain the desired performance over the range of operating conditions. Airfoil shape optimization over a range of Mach numbers was considered for constant lift-to-drag ratio by minimizing a weighted integral over the design operating envelope. Through a numerical integration scheme, the weighted integral is converted to a multi-point optimization problem with specified sampling points and weights. In more recent studies, Epstein *et al.* [2009] adopted the weighted sum strategy for three-dimensional wing drag optimization. The authors performed a comprehensive comparative study of different optimization techniques, namely an adjoint-based GM, a response surface method and a GA, in an effort to construct a reliable test case-study for aerodynamic design optimization frameworks. In the recent study by Buckley *et al.* [2010], the scope of multi-point aerodynamic optimization has been extended to accommodate the full range of flight conditions that an aircraft encounters during a mission, characterized by eighteen design points.

Despite its popularity among researchers, regardless of the optimizer, there are recognized deficiencies with the weighted sum strategy [see Marler and Arora, 2004, and references therein]. The primary disadvantage of the weighted sum strategy is the *a priori* selection of weights does not necessarily guarantee that the final solution will reflect the preferred interests of the designer, and it may be necessary to resolve the problem with new weights [Messac, 1996]. The second deficiency is that as a consequence of aggregating the objective functions, it is impossible to obtain solutions on non-convex Pareto fronts[9] [Deb, 2001; Marler and Arora, 2004]. Another disadvantage of the weighted sum method is that varying the weights continuously will not necessarily result in an even distribution of Pareto optimal solutions [Das and Dennis, 1997; Deb, 2001; Marler and Arora, 2004], nor a complete representation of the Pareto front [Arora, 2004].

[9] See [Das and Dennis, 1997; Messac *et al.*, 2000a;b] and [Jin *et al.*, 2001] for discussions on non-convex Pareto fronts

3.5.2 Evolutionary Multi-objective Optimization (EMO)

While EA could potentially be used as an optimizer for the *a priori* methods discussed previously, their appeal is in their ability to converge to the whole Pareto front of a given problem, thus identifying numerous trade-off solutions [Marler and Arora, 2004]. The designer therefore has the flexibility to select the most appropriate solution(s) which best suit the intended application. GM techniques are not ideal for MOO, since only one candidate is progressed through the optimization loop and consequently only one final trade-off can be identified. In order to generate a spread of solutions, it would therefore be necessary to perform consecutive runs of the optimization loop. Alternatively the use of evolutionary methods is ideal, since they maintain a population of candidates capable of identifying numerous trade-offs simultaneously. As confirmed by Coello Coello [2007],

> *"...from the several emergent research areas in which (evolutionary algorithms) have become increasingly popular, multi-objective optimization has had one of the fastest growing in recent years... Researchers and practitioners remain to find an irresistible match between the population available... and the need in multi-objective problems to approximate the Pareto trade-off curve or surface."*

Numerous texts are available (and are continuously updated), which are devoted entirely to Evolutionary Multi-objective Optimization (EMO) [for example Fonseca and Fleming, 1995; Deb, 2001; Coello Coello, 2007]. Monitoring the frequency that these texts are cited in the literature illustrates the popularity of these techniques among researchers[10]. It is therefore intractable that one could hope to concisely present a review of EMO within the confines of a thesis report. The field of EMO is extensive, and while the reported algorithms differ in their own respects, they generally maintain a population of candidates which is iteratively updated through the dominance criterion. The use of other EMO methods which do not make explicit reference to the dominance criteria have also been reported. For example the vector evaluated GA proposed by Schaffer [1989] or the use of preference-based mechanisms to identify optimal trade-offs in many-objective (i.e. $m > 4$) environments [see Wickramasinghe and Li, 2009]. The present discussion is therefore limited to *some* of the dominance-based EMO techniques for aerodynamic design, or concepts which have prompted the development of frameworks tailored for aerodynamic design.

The primary difficulty in developing EMO algorithms is how to quantify the success of a solution among other candidates, and therefore how to iteratively update the population [Marler and Arora, 2004]. Another issue arises from the need to maintain diversity

[10] A list of references on EMO is also maintained: `http://www.lania.mx/ ccoello/EMOO/EMOObib.html`

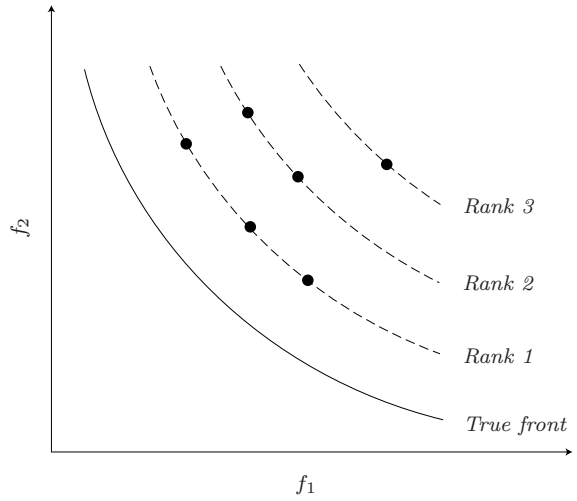


Figure 3.11: Illustration of non-dominated sorting, where identified solutions are ranked according to their pertaining front and closeness to the true Pareto front

in the non-dominated set, and maintain already identified non-dominated solutions[11]. An initial attempt at extending GA to handle multiple objectives was by Goldberg [1989]. At each generation individuals are assigned a rank, whereby non-dominated solutions have the highest rank and solutions in the most inferior front have the lowest rank. This ensures that the more successful individuals are selected to reproduce. Fonseca and Fleming [1993] proposed a similar ranking procedure to Goldberg. In their Multi-Objective Genetic Algorithm (MOGA) each individual is assigned a rank based on the number of individuals that dominate them. A niching method is implemented in order to maintain a diverse set of non-dominated solutions in the Pareto front. Srinivas and Deb [1994] later extended the approach by Goldberg, with the Non-dominated Sorting Genetic Algorithm (NSGA). All individuals are ranked based on non-domination, and correspondingly the highest selection pressure for reproduction is given to solutions in the non-dominated front, and so forth until all individuals are classified. An extension to the NSGA algorithm (NSGA-II algorithm) was developed by Deb *et al.* [2002]. This algorithm introduces elitism and a crowding distance operator, providing more efficient convergence rates over its predecessor. An illustration of non-dominated sorting is provided in Figure 3.11.

Due to the rate at which new algorithms and concepts were being developed, EMO have gained popularity among researchers in aerodynamic design. The concept of applying EMO for aerodynamic design was popularized by the pioneering work of Obayashi *et al.*. MOGA was applied for the aerodynamic design of transonic wings [1997], the shape opti-

[11] The final issue – perhaps more related to their application in engineering design – is the computational cost associated in maintaining a population of candidate solutions. It can be argued that the introduction of the parallel computing architecture has somewhat alleviated this issue [Keane and Nair, 2005]

mization of cascade airfoils [2000a] as well as supersonic wing-shape optimization [2000b]. These studies were fairly complex, yet the advantages of driving a population of candidates to identify trade-offs were comprehensively highlighted. It was also one of the first studies where EMO was applied for case-studies of varying fidelity, illustrating its proficiency in handling different multi-objective environments. Crossley *et al.* [1996; 1998] report the use of a MOGA variant, which was further developed and applied to the aerodynamic and aeroacoustic design of airfoils by Jones *et al.* [1998]. A MOGA was also developed and applied by Vicini and Quagliarella [1997] for the aerodynamic design of wings. The algorithm was further developed by introducing a GM in the form of a genetic operator to increase convergence rates [1999], and later applied to the shape optimization of both transonic airfoils and multi-component airfoils for high-lift applications [Quagliarella and Vicini, 2001]. The NSGA algorithm [Srinivas and Deb, 1994] is applied in the airfoil aerodynamic design framework of Marco *et al.* [1999]. Ray *et al.* [2001] also applied the NSGA algorithm for engineering design applications, before developing a multi-objective swarm algorithm for airfoil shape optimization [2002; 2004].

Most of the earlier attempts at developing EMO algorithms were derived using the principles of GA. However Pareto optimality is not linked with any of the conventional genetic operators. Consequently, it is possible that a Pareto optimal solution may be identified and then randomly discarded [Marler and Arora, 2004]. For this reason, the use of elitist schemes has also become common [Marler and Arora, 2004; Coello Coello, 2007]. Elitism generally consists of the use of an archive, that is used to store identified non-dominated solutions and can also interact with the population of individuals. This has also naturally led to the development of multi-objective algorithms based on other EA techniques. The Pareto Archived Evolution Strategy (PAES) by Knowles and Corne [2000] is one such strategy, whereby evolutionary operations and an archive of non-dominated solutions are used to compare with individuals in the current population. This led to the development of the earliest published attempt at extending particle swarm optimization to handle multiple objectives by Coello Coello and Lechuga [2002]. There has since been an increase in the number of proposals for multi-objective particle swarm optimization [Engelbrecht, 2005; Reyez-Sierra and Coello Coello, 2006]. This research focuses on the development of a novel swarm-based framework for multi-objective aerodynamic design.

3.5.3 Multi-objective Particle Swarm Optimization (MOPSO)

PSO is an effective tool for single-objective optimization problems due to its fast convergence [Trelea, 2003]. It has also gained rapid popularity for MOO [Reyez-Sierra and Coello Coello, 2006]. PSO is a population-based technique, thus it can be tailored to

identify trade-off solutions to a MOP in one single run, similar to EMO. Comprehensive surveys on extending PSO to handle multiple objectives are provided by Engelbrecht [2005]; Reyez-Sierra and Coello Coello [2006]. It was established that the primary ambiguity in specifically tailoring PSO to handle multiple objectives was the selection of guides for each particle to avoid convergence to a single solution. The velocity update Eq. (3.5) makes explicit reference to a *single* personal best and global best solution. The selection process for particle leaders must therefore be restructured, to encourage search diversity and ensure non-dominated solutions found during the search are maintained.

Initial attempts at a Multi-Objective Particle Swarm Optimization (MOPSO) algorithm were motivated by the archive strategy by Knowles and Corne [2000]. Coello Coello and Lechuga [2002] incorporated the concept of Pareto dominance in PSO by maintaining two independent populations: the particle swarm and the elitist archive. Non-dominated solutions are stored in the archive and subsequently used as neighbourhood leaders. The objective space is separated into hypercubes, which serve as a particle anti-clustering mechanism. Solutions in sparsely populated hypercubes have a higher selection pressure to be leaders and solutions in densely populated hypercubes are removed if the archive limit is exceeded. The initial approach by Coello Coello and Lechuga was later extended by Mostaghim and Teich [2003], where the concept of ϵ-dominance[12] was studied and compared to existing clustering techniques for fixing the archive size, with favourable results. Ray and Liew [2002] also developed an efficient evolutionary-based swarm paradigm, where a Pareto ranking scheme similar to non-dominated sorting is implemented to rank each solution in terms of the objective functions and constraint violation.

Fieldsend and Singh [2002] addressed the computational complexity of maintaining a restricted archive, by incorporating the *dominated tree* method. This data structure allows for an unrestricted archive size, which interacts with the population to define global leaders. A *turbulence* operator (similar to the concept of mutation in EA) was also implemented, where swarm members were randomly displaced on the design space to reduce the probability of premature stagnation. In the NSPSO algorithm of Li [2003], the non-dominated sorting mechanisms of NSGA-II are incorporated. The population and the personal best position of each particle are consolidated to form one single population, and the non-dominated sorting scheme is utilized to rank each solution. Global guides are selected based on particle clustering, where a niching or crowding distance metric is used to further classify non-dominated solutions. Li later proposed the maximinPSO algorithm [Li, 2004], which uses no niching method to maintain diversity.

[12] Refer to Deb [2001] for a description of ϵ-dominance

Reyez-Sierra and Coello Coello [2005] proposed an elitist archive incorporating the ϵ-dominance strategy to maintain global leaders for the swarm. A crowding distance operator is employed to classify non-dominated solutions and maintain uniformity. The crowding distance operator is also used to limit the number of candidate leaders after each population update, simplifying the mechanism to control the set of candidate leaders. A turbulence operator is implemented to encourage diversity, whereby particles are randomly mutated. A similar approach by Raquel and Naval [2005] was developed in parallel (although this method does not implement ϵ-dominance), where the crowding distance was used to both define the global guides and truncate the size of the archive. For this research, a MOPSO variant is proposed (described in Chapter 5) which is predominately influenced by the two latter studies.

3.5.4 Preference-based Optimization

The *a priori* strategies (such as the weighted sum approach) articulate preferences by initially prioritizing each objective. These methods are simple and efficient if the fitness function is able to faithfully reflect the preferred interests of the designer. However, this is rarely the case since there is no prior knowledge of the relative influence of each objective in order to allocate bias with sufficient confidence [Rachmawati and Srinivasan, 2006]. The dominance-based EMO approach alternatively provides the designer greater flexibility in selecting the most preferred solution. However it poses the challenge of identifying and exploiting the entire Pareto front which may be impractical for design applications due to the excessive number of function evaluations. While conventional EMO techniques may be computationally demanding, Fonseca and Fleming [1995] argue that their most attractive aspect is the intermediate information generated which can be exploited by the designer to refine preferences and improve convergence. This concept of interactive optimization has led to an increasing interest in coupling classical *a priori* methods to EMO as an intuitive way of specifying designer preferences and identifying solutions of interest. This is known as *preference-based* optimization, which is the motivation for this research.

Comprehensive surveys on preference-based optimization are provided by Coello Coello [2000] and Rachmawati and Srinivasan [2006]. The first recorded attempt at incorporating preferences within an EMO framework was by Fonseca and Fleming [1993] using the goal programming approach. Goal programming [see Marler and Arora, 2004] is an ideal approach to indicate desired levels of performance for each objective, since they are closely related to the final solution of the problem. Goals may either represent target or ideal values. Fonseca and Fleming later extended the approach where an online decision making strategy was proposed based on goal and priority information [1998]. A goal programming

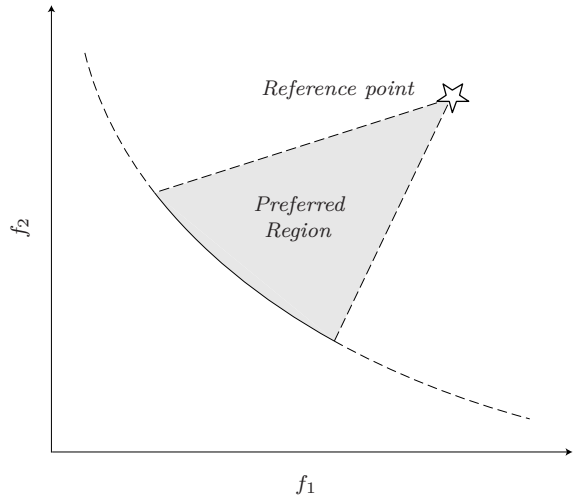


Figure 3.12: Isolating the preferred region using the reference point compromise

mechanism for identifying preferred solutions for MOP was also proposed by Deb [1999]. While the reported frameworks draw on the preferred interests of the designer to aid the optimization process, the goal programming approach is computationally complex, and there is no means of specifying any relation or trade-off between the objectives [Rachmawati and Srinivasan, 2006].

Thiele *et al.* [2007] proposed another variant of preference-based EMO. A coarse representation of the Pareto front is initially presented to the designer. The most interesting regions are subsequently isolated, to which the algorithm continues to focus on exclusively. This proposal effectively removes the necessity to predefine target values for each objective and provides the designer a means of isolating the preferred trade-offs. However, it is a two-stage approach requiring an initial approximation to the Pareto front, which may be unnecessarily expensive. The integration of other classical preference articulation methods have also been proposed in the literature. A reference point based EMO framework was proposed by Deb and Sundar [2006]. The crowding distance operator of the NSGA-II algorithm was modified to include the reference point information and the extent of the preferred region was controlled by ϵ-dominance. Figure 3.12 demonstrates the concept of the preferred region based on the stipulation of a reference point compromise. Deb and Kumar also experimented the use of other classical preference methods, such as the reference direction method [2007a] and the light beam search method [2007b].

Recently the concept of preference-based optimization has also been integrated within MOPSO frameworks. Wickramasinghe and Li [2008] integrated the reference point method to both the NSPSO and maximinPSO algorithms of Li [2003; 2004]. Significant improvement in convergence efficiency was highlighted and it was demonstrated that final solutions

are of higher relevance to the designer. Wickramasinghe and Li [2009] later extended their approach to handle many-objective problems, by replacing the dominance criteria entirely with the simpler distance metric. The benefits of utilizing the reference point method for a many-objective aerodynamic optimization framework was also investigated [2010]. It was conclusively demonstrated that without the use of the reference point, obtaining a final set of preferred designs solely through conventional EMO techniques is improbable. The proposed multi-objective algorithm (refer to Chapter 5) is primarily influenced by the preference-based architecture developed by Wickramasinghe and Li.

3.6 Summary

In this chapter the optimization concepts which are related to this research have been introduced. A generic discussion on the characteristics of the optimization problem, such as the definition of global and local optima has been provided. This is followed by a comparison of the two main optimization techniques, namely the deterministic gradient-based methods and the stochastic evolutionary algorithms. A comprehensive introduction and review of the evolutionary algorithm selected for this research, known as particle swarm optimization, is subsequently provided. This research is concerned primarily with multi-objective optimization, and as such a thorough introduction to dominance and the techniques utilized to identify non-dominated solutions has been given. Focus is given to the field of preference-based optimization, since it is the primary theme of the proposed design framework which is documented in Chapter 5.

Chapter 4

Surrogate Modelling

The most prohibiting factor of design optimization is the cost of evaluating the objective and constraint functions. For high-fidelity aerodynamic design, function evaluations may very well be measured in hours. This computational burden ultimately questions the practicality of performing an optimization study, and is often alleviated by simply reducing the level of sophistication of the solver. This consequently reduces the fidelity of the final design, which is undesirable. Another mitigating strategy which has steadily gained popularity in design is the use of inexpensive surrogates or metamodels. These models emulate the response of the expensive function at an unobserved location, based on observations at other locations. Surrogate models are not specifically optimization methods, but rather they may ideally be used in lieu of the expensive function to extract information from or visualize the design space during the optimization process. This chapter presents a brief introduction to surrogate modelling, with a comprehensive review on the Kriging method.

4.1 Surrogates in Design

The insightful texts by Keane and Nair [2005] and Forrester *et al.* [2008] provide a detailed account of the use of surrogates in design. The most common use is to construct a *curve fit* of an expensive function landscape which can be used to predict results without recourse to the original function. This is supported by the assumption that the inexpensive surrogate will still be usefully accurate when predicting sufficiently far from observed data points [Forrester *et al.*, 2008]. Figure 4.1 illustrates the use of a surrogate to fit the one-dimensional multi-modal function, based on four sample observations. The original function landscape could potentially represent any deterministic quantity of the design space. Rather than exactly emulating the response of a high-fidelity flow solver, the surrogate may be used to bridge the gap between flow solvers of varying fidelity [Forrester

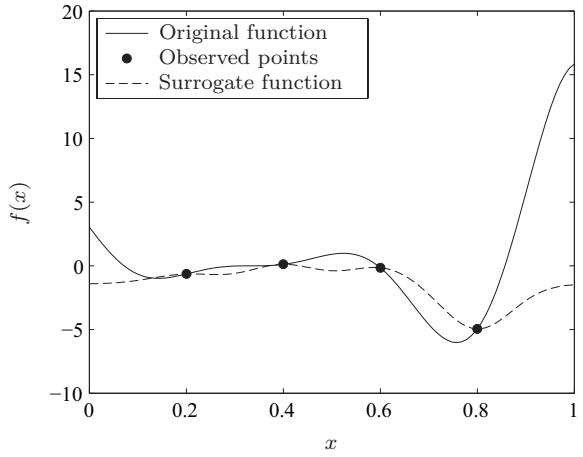


Figure 4.1: Constructing a surrogate to fit a one-dimensional function

et al., 2008]. Alternatively a surrogate may be used to filter noisy landscapes, as to eliminate the adverse effects of flow solver convergence or grid discretization. Surrogates are also used for data mining. Such methodologies are applied to extract useful information on the relationship between the design space and the objective space, allowing informed decisions to be made which could simplify a seemingly complex problem.

4.1.1 Constructing a Surrogate

For the aforementioned uses of surrogate modelling, the common requirement is to replicate the function relationship between the variable inputs and the output quantity of interest. This is typically achieved by sampling the design space using the exact function to sufficiently model the underlying relationship within the allowable computational budget. Whether the aim is to locally model the design space surrounding an existing design or train a surrogate to replicate the global design space is almost entirely dependent on the formation of the sampling plan [Keane and Nair, 2005]. The construction of a surrogate model in either case should, wherever possible, ideally make use of a parallel computing structure. A suitable surrogate model $\hat{f}$ of the precise objective function f should then be constructed to fit the dataset.

There are a multitude of popular techniques for constructing surrogates in the literature. For a comprehensive review of different methods, the reader is referred to (among others) Jones [2001]; Jin [2005]; Keane and Nair [2005]; Forrester *et al.* [2008]. The selection of the surrogate model is dependent on the information in which the designer is attempting to extract from the design space. Polynomial response surfaces and radial basis functions are a fairly popular technique for constructing local surrogates, especially if some level of regression is desirable. Techniques such as Kriging or support vector

machines are more ideally suited to global optimization studies since they offer greater flexibility in tuning model parameters and provide a confidence interval of the predicted output. Neural networks require extensive training and validation, yet have also been a popular technique for design applications notably in aerodynamic modelling [Greenman and Roth, 1999] and visualization techniques [see Kohonen, 1995; Jeong *et al.*, 2005b].

4.1.2 Managing Surrogates in Evolutionary Optimization

Despite the benefits of evolutionary computing, their application in solving aerodynamic design optimization problems is computationally challenging. Navigating a noisy or multi-modal design landscape with a high-fidelity objective flow solver often requires an excessive number of fitness evaluations before an acceptable solution is identified. The concept of incorporating surrogate modelling in optimization is motivated by the need to alleviate this computational burden. Comprehensive surveys on incorporating surrogate modelling within the evolutionary optimization architecture are provided by Jin [2005] and Keane and Nair [2005]. This section focuses on the generic structure of the surrogate-based optimization architecture which involves:

1. Generating the sampling plan to create the dataset $\mathbf{X} = \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^N\}^T$ and the corresponding response $\mathbf{Y} = \{f(\mathbf{x}^1), f(\mathbf{x}^2), \ldots, f(\mathbf{x}^N)\}^T$.

2. Constructing the surrogate model $\hat{f}(\mathbf{x})$ based on the dataset $[\mathbf{X}, \mathbf{Y}]$.

3. Searching the surrogate $\hat{f}(\mathbf{x})$ using the optimizer and identify promising solutions.

4. Verifying the identified solutions by calculating the exact response value using $f(\mathbf{x})$.

5. Checking for convergence else augment the dataset $[\mathbf{X}, \mathbf{Y}]$ with the additional solutions from step 4 and repeat from step 2.

The surrogate model is constantly re-constructed, whenever the dataset is augmented with an additional solution. The optimization algorithm then explores or exploits the surrogate function to identify promising areas of the design space. Where the principal ambiguity lies is the verification or co-operation between the exact function $f(\mathbf{x})$ and the surrogate model $\hat{f}(\mathbf{x})$ during the optimization process. While the aim of the surrogate-based optimization architecture is to significantly reduce the number of precise function evaluations, it must also be ensured that the optimizer converges to the near global optimum (or trade-offs) of the original fitness function. The excellent text by Jin [2005] outlines the main approaches to managing surrogate models for evolutionary frameworks:

No Control: The surrogate is assumed to be globally accurate and the fidelity of the model is sufficient such that it can replace the exact function entirely. This technique is a rarity for aerodynamic optimization studies, since (depending on the function landscape) it would require an unnecessarily large sampling plan.

Fixed Control: In this strategy the surrogate and the exact function are used co-operatively. The surrogate model can either be managed through generation- or individual-based control. For generation-based control, the surrogate is initially assumed to be globally accurate and the evolutionary optimizer operates on the surrogate model. After a fixed number of generations, the surrogate is updated with the identified solutions. An example of this strategy is provided in Chung and Alonso [2004] for the design of low-boom supersonic business jets. The non-dominated trade-offs identified from exploring the surrogate after a fixed number of generations were used to update and verify the prediction accuracy of the surrogate model until convergence. For individual-based control, the fitness of most individuals in the population is computed using the surrogate, while a subset of individuals use the exact objective function. The population subset contains the best performing solutions of the current generation. This strategy is used by Jeong *et al.* [2005a]; Kanazaki *et al.* [2006], where at each generation the solutions which are expected to provide the greatest improvement in any objective are used to update the surrogate. Individual-based control is a popular strategy for aerodynamic design frameworks and has also been considered in Jin *et al.* [2002]; Ong *et al.* [2004]; Emmerich *et al.* [2006]. However, as highlighted by Jin [2005], the drawback of these methods is that the frequency of updating the surrogate is fixed. This is not practical since the model fidelity may vary significantly during the optimization process.

Adaptive Control: This strategy is similar to the fixed control strategy, with the exception that the frequency of surrogate updates is not dependent on a population subset or a fixed number of generations but rather on the fidelity of the surrogate model. Since the fidelity of the surrogate model is likely to fluctuate during the optimization process, individuals are selected for exact evaluation based on a screening criterion, which quantifies the prediction accuracy of the surrogate based on a confidence output. Within an adaptive control framework, it is therefore common that the population subset selected for exact evaluation may be larger during the exploration phase and subsequently reduce when the search begins to exploit promising areas of the design space. Examples of the adaptive control strategy are given in the design frameworks of D'Angelo and Minisci [2005]; Li *et al.* [2008] where error-based screening criteria are used to filter trustworthy solutions.

For high-fidelity aerodynamic design, the use of surrogate modelling is a necessity to maintain a tractable computational complexity. This research proposes a novel adaptive control strategy for Kriging modelling based on the concept of adaptive control.

4.2 Sampling Arrays

Construction of a globally accurate surrogate which exactly replicates the response of an aerodynamic function landscape is improbable due to the multi-modal design space. It is more common to construct a global surrogate approximation based on a limited number of observations and then locally update the prediction accuracy of the surrogate as the search progresses towards promising areas of the design space. The spatial arrangement of these initial observations is governed by the sampling array $\mathbf{X}$ of size (N_K, n) and the corresponding response $\mathbf{Y}$ of size $(N_K, 1)$ such that,

$$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{N_K}\}^T \quad \text{and} \quad \mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{N_K}\}^T \tag{4.1}$$

The fidelity of the surrogate model has a large dependence on the sampling plan, and the number of samples N_K is also dependent on the dimensionality of the problem. If a certain level of prediction accuracy is achieved by sampling a one-parameter design space in N_K locations, then to achieve the same full factorial sample in a n-dimensional space, N_K^n observations are required. This is termed the *curse of dimensionality* and poses a large problem for high-dimensional design problems [Forrester *et al.*, 2008; Sóbester, 2009]. It is therefore imperative that the number of sample points required to generate a fairly accurate approximation to the response function is minimized [Forrester *et al.*, 2008].

4.2.1 Sampling Techniques

The excellent texts by Keane and Nair [2005] and Forrester *et al.* [2008] provide a comprehensive discussion on sampling methods, in the context of constructing surrogates for design problems. Classical sampling techniques such as the full factorial design were developed to alleviate the adverse effects of measurement noise in physical experiments [Montgomery, 1997]. These classical designs provide a simple way of producing a uniform sample of the design space through rectangular grids. However to evaluate every possible permutation for the n-dimensional design space is computationally challenging. While classical designs are stratified over the unit hypercube, design permutations are biased in each dimension, which is a consequence of enforcing uniformity through rectangular grids. Unlike physical experiments, computer experiments are not influenced by random noise. Therefore to model the underlying functional relationship between variables inputs and the function response, it is imperative that the design space is sampled optimally.

Developing a good sampling plan is an optimization problem. It is desirable to generate a sampling plan with the fewest evaluations, whilst ensuring the entire design space is sufficiently sampled. The strategy here is to therefore provide a stratified or unbiased representation of the design space rather than attempting to establish a truly accurate representation. Notable techniques include Monte Carlo sampling [Sobol, 1994], Latin hypercube sampling [Mckay *et al.*, 1979] and orthogonal arrays [Hedayat *et al.*, 1999].

4.2.2 Latin Hypercube Design

Latin hypercube sampling (LHS) is a statistical method first described by Mckay *et al.* [1979]. LHS generates space-filling designs using a relatively small number of design points to sample a multi-dimensional space. A stratified sample plan is constructed in the context of a Latin square, where sample positions appear only once in every row and every column. Figure 4.2 illustrates an LHS design for an $n = 3$ grid of $N_K = 10$ samples.

Figure 4.2 demonstrates that an LHS design will generate a random stratified sampling plan, where the projections on to the axes are uniformly spread. This concept does not however guarantee that the plan will be ideal, since a stratified sample plan is also possible by placing all points on the main diagonal, which will result in a crude approximation of the function landscape [Forrester *et al.*, 2008]. Additional criteria are therefore required to ensure stratified designs which effectively sample a large percentage of the n-dimensional design space. Techniques to develop space-filling LHS designs were pioneered by the work of Johnson *et al.* [1990], and Morris and Mitchell [1995] through their development and application of the *maximin* space-filling criterion. These designs attempt to maximize the minimum distance between sample points. Generating optimal designs is still an active area of research today particularly in the field of computational statistics and data analysis Santer *et al.* [2003]. In this research space-filling designs are generated by reducing the correlation between sample points, as discussed in detail by Forrester *et al.* [2008]. The correlation criterion is optimized to generate a space-filling design within the computational budget of N_K samples.

4.3 Kriging Method

A popular surrogate method which is ideally suited to optimization is based on Gaussian stochastic process models. This method was originally developed in the field of geostatistics and is referred to as Kriging[1]. In the optimization literature, Kriging was popularized

[1]The term *Kriging* is in honour of the South African mining engineer D. G. Krige, who first proposed the method [see Matheron, 1963]

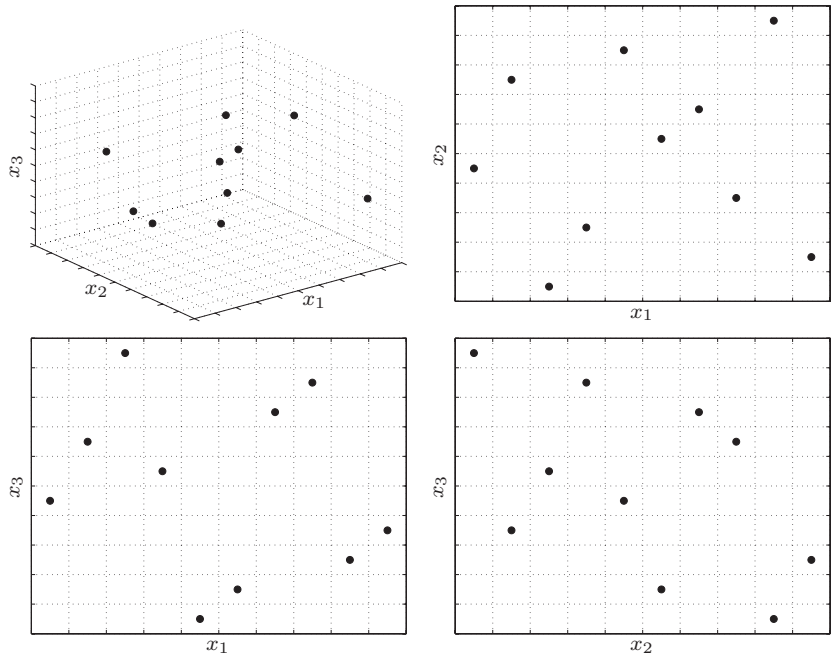


Figure 4.2: Illustration of LHS design for a $n = 3$ dimensional grid of $N_K = 10$ samples.

by the work of Sacks *et al.* [1989] and is also referred to as the Design and Analysis of Computer Experiments (DACE). The derivation of the Kriging method is available from a number of sources in the literature, and the readers are encouraged to follow the excellent texts by Jones [2001] and Forrester *et al.* [2008]. This section is devoted to the techniques for the construction and validation of Kriging models, and introduces concepts essential for their integration to the proposed optimization framework.

4.3.1 Model Construction

The Kriging method predicts the response of a function at an unobserved location, based on responses at prior observations. Given the sampling array $[\mathbf{X}, \mathbf{Y}]$, the ordinary Kriging method expresses the unknown function $y(\mathbf{x})$ as,

$$y(\mathbf{x}) = \beta + z(\mathbf{x}). \tag{4.2}$$

Where $\mathbf{x} = (x_1, \cdots, x_n)$ is the data location, β is a constant global mean value and $z(\mathbf{x})$ represents a local deviation at the data location $\mathbf{x}$ based on a stochastic process with zero-mean and variance σ^2 following the Gaussian distribution. The stochastic variables of any two locations $\mathbf{x}^i$ and $\mathbf{x}^j$, are correlated with each other using the Gaussian basis function expression

$$R(\mathbf{x}^i, \mathbf{x}^j) = \exp\left(-\sum_{k=1}^{n} \theta_k |x_k^i - x_k^j|^{p_j}\right), \tag{4.3}$$

where R is the correlation matrix. It is now possible to construct the $N_K \times N_K$ correlation matrix of all the observed data:

$$\mathbf{R} = \begin{pmatrix} R(\mathbf{x}^1, \mathbf{x}^1) & \cdots & R(\mathbf{x}^1, \mathbf{x}^{N_K}) \\ \vdots & \ddots & \vdots \\ R(\mathbf{x}^{N_K}, \mathbf{x}^1) & \cdots & R(\mathbf{x}^{N_K}, \mathbf{x}^{N_K}) \end{pmatrix} \tag{4.4}$$

The correlation between the sample data expects that the data will be smooth and continuous, which is the principle assumption of the Kriging method. The data in the sample $\mathbf{Y}$ is correlated through the matrix $\mathbf{R}$. The correlation between any two sample points is shown to depend entirely on the absolute distance between the points $|\mathbf{x}^i - \mathbf{x}^j|$ and the correlation parameters $\mathbf{p}$ and $\boldsymbol{\theta}$.

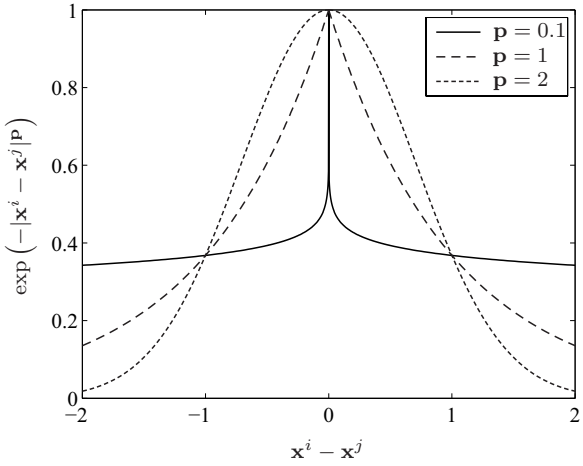
4.3.2 Model Training

It is observed that the correlation between sample points, whilst being dependent on the sampling plan, is also dependent on the parameters $\mathbf{p}$ and $\boldsymbol{\theta}$. There are no globally intuitive values for these parameters, and optimum values should be selected in order to maximize the prediction accuracy or *train* the Kriging model to fit a given sample dataset.

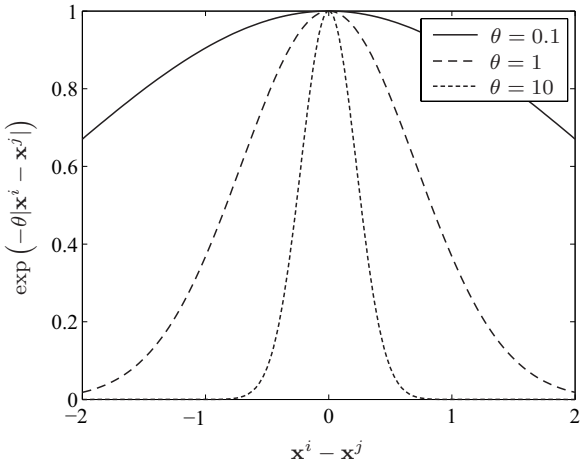
Correlation Parameters

The effect of the correlation parameter $\mathbf{p}$ is shown in Figure 4.3(a). It is observed that the correlation between any two points approaches unity as the sample points coincide, i.e. $\exp\left(-|\mathbf{x}^i - \mathbf{x}^j|^{\mathbf{p}}\right) \to 1$ as $\mathbf{x}^i - \mathbf{x}^j \to 0$. This is analogous to stating that $y(\mathbf{x}^i)$ is equal to $y(\mathbf{x}^j)$. Similarly, it is observed that as $\mathbf{x}^i - \mathbf{x}^j \to \infty$, the correlation $\exp\left(-|\mathbf{x}^i - \mathbf{x}^j|^{\mathbf{p}}\right) \to 0$. The rate of correlation is shown to vary with the parameter $\mathbf{p}$. Essentially, larger values of $\mathbf{p}$ provide a smooth continuous gradient through the point $\mathbf{x}^i - \mathbf{x}^j = 0$. However, this is linked with a rather rapid decline in the correlation as the distance between sample points increases. With lower values of $\mathbf{p}$, it is observed that there is a fairly constant correlation between any two sample points and there is a near discontinuity at the point $\mathbf{x}^i - \mathbf{x}^j = 0$, which is counter intuitive. For this reason, Jones [1998] suggests a range of $1 \leq \mathbf{p} \leq 2$, with many researchers opting for a constant $\mathbf{p} = 2$ [Emmerich *et al.*, 2006; Gano *et al.*, 2006; Jin, 2005; Jeong *et al.*, 2005a; Li *et al.*, 2008; Lee *et al.*, 2005].

The effect of $\boldsymbol{\theta}$ is shown in Figure 4.3(b). This parameter does not affect the smoothness of the correlation, yet it is inversely proportional to the extent of the influence of a given sample point. It is demonstrated that a lower value of $\boldsymbol{\theta}$ suggests that sample points have a high correlation and that the fitness response is fairly similar. A higher value of $\boldsymbol{\theta}$ indicates that the fitness response is rapidly changing. As discussed by Jones [1998] and



(a) Varying $\mathbf{p}$



(b) Varying $\boldsymbol{\theta}$

Figure 4.3: Change in sample point correlation due to varying parameters

Forrester *et al.* [2008], in multi-dimensional problems the value of θ_j can thus be viewed as a measure of importance or activity of the design variable x_j. If the design variable x_j has no apparent influence on the fitness response, then it would be expected that θ_j should be close to zero. On the other hand, if the variable x_j has a significant influence on the fitness response, then a larger value of θ_j is expected.

Maximum Likelihood Estimation

The parameters $\mathbf{p}$ and $\boldsymbol{\theta}$ have a significant impact on sample point correlation. It was also shown that the parameter $\boldsymbol{\theta}$ provides a measure of the level of activity of each design variable. In the literature, it is observed that the most popular method of estimating the values of $\mathbf{p}$ and $\boldsymbol{\theta}$ is to maximize the likelihood of $\mathbf{Y}$. Maximizing the likelihood of $\mathbf{Y}$ is an attempt to reduce the errors obtained by constructing a surrogate to exactly replicate

the precise function. Following the derivation of Jones [2001], the likelihood expressed in terms of the sample data is,

$$L = \frac{1}{(2\pi)^{\frac{N_K}{2}}(\sigma^2)^{\frac{N_K}{2}}|\mathbf{R}|^{\frac{1}{2}}} \exp\left[\frac{-(\mathbf{Y}-\mathbf{1}\beta)^T\mathbf{R}^{-1}(\mathbf{Y}-\mathbf{1}\beta)}{2\sigma^2}\right]. \tag{4.5}$$

It is more convenient to express the natural log of the likelihood function,

$$\ln L = -\frac{N_K}{2}\ln(2\pi) - \frac{N_K}{2}\ln(\sigma^2) - \frac{1}{2}\ln(|\mathbf{R}|) - \frac{(\mathbf{Y}-\mathbf{1}\beta)^T\mathbf{R}^{-1}(\mathbf{Y}-\mathbf{1}\beta)}{2\sigma^2}. \tag{4.6}$$

Equating the derivatives of the log likelihood function with respect to β and σ^2 to zero, the maximum likelihood estimates for the mean and standard deviation are obtained:

$$\hat{\beta} = \frac{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{Y}}{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}}, \tag{4.7}$$

$$\hat{\sigma}^2 = \frac{1}{N_K}\left[(\mathbf{Y}-\mathbf{1}\hat{\beta})^T\mathbf{R}^{-1}(\mathbf{Y}-\mathbf{1}\hat{\beta})\right]. \tag{4.8}$$

Substituting Eq. (4.7) and (4.8) back into Eq. (4.6), the concentrated natural log likelihood function is obtained. Ignoring all constants, the concentrated ln-likelihood function is

$$\ln(L) \approx -\frac{N_K\ln(\hat{\sigma}^2) - \ln|\mathbf{R}|}{2}. \tag{4.9}$$

The value of the concentrated ln-likelihood function is dependent on the value of the correlation matrix $\mathbf{R}$ and hence for a given sample dataset is solely dependent on the correlation parameters $\mathbf{p}$ and $\boldsymbol{\theta}$. Furthermore, if a constant value of $\mathbf{p} = 2$ is considered, determining the maximum likelihood is reduced to an n-variable optimization problem,

$$\min \quad \frac{N_K\ln(\hat{\sigma}^2) - \ln|\mathbf{R}|}{2} \quad \text{subject to: } \theta_j > 0, \quad \forall j = 1,\ldots,n \tag{4.10}$$

Training a Kriging model from a sample dataset is time-consuming and is $\mathcal{O}(N_K^3)$. To visualize an example of the ln-likelihood function landscape, the two-variable *Branin* test function is considered (see Appendix A). The variables are normalized to the range $[0, 1]$ for ease of comparison. A LHS of 20 points is used to construct the Kriging model. The variable $\boldsymbol{\theta}$ is converted to a logarithmic scale, as suggested by Forrester *et al.* [2008]. The concentrated ln-likelihood function landscape is visualized in Figure 4.4.

The concentrated ln-likelihood function landscape is difficult to navigate since the function is multi-modal and is susceptible to long flat ridges. Gano *et al.* [2006] conducted a series of tests where a number of deterministic and stochastic optimizers where compared for two engineering design problems. It was concluded that although local optimization tools such as Quasi-Newton or pattern search may prematurely terminate the search due to convergence to local minima, they are still economically more efficient than stochastic algorithms (such as EA). Stochastic optimizers potentially overcome the multi-modality of the design space, but at the expense of many orders of magnitude more evaluations.

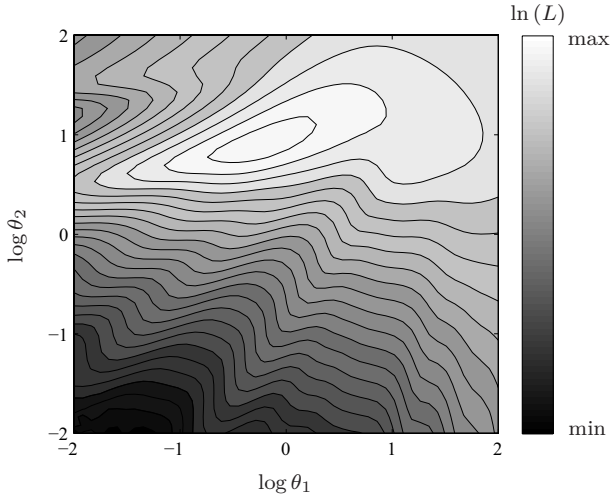


Figure 4.4: Concentrated ln-likelihood function landscape for *Branin* test function

4.3.3 Model Prediction

The correlated sample data is then used by the Kriging model to make predictions at unobserved locations. The following derivation is extracted from Jones [2001]. The Kriging correlation parameter $\boldsymbol{\theta}$ is optimized such that it maximizes the likelihood of the observed data, $\mathbf{Y}$. Given the correlation parameter, a new prediction $\hat{y}$ at the position $\mathbf{x}$ is chosen such that the likelihood of the sample data *and* the prediction is maximized. To achieve this a vector of correlations is defined, augmenting the observed dataset $\mathbf{Y}$ with the still to be determined prediction $\hat{y}$,

$$\mathbf{r} = \begin{pmatrix} R(\mathbf{x}^1, \mathbf{x}) \\ \vdots \\ R(\mathbf{x}^{N_K}, \mathbf{x}) \end{pmatrix}. \tag{4.11}$$

The correlation matrix for the augmented dataset $\tilde{\mathbf{R}}$ is given as

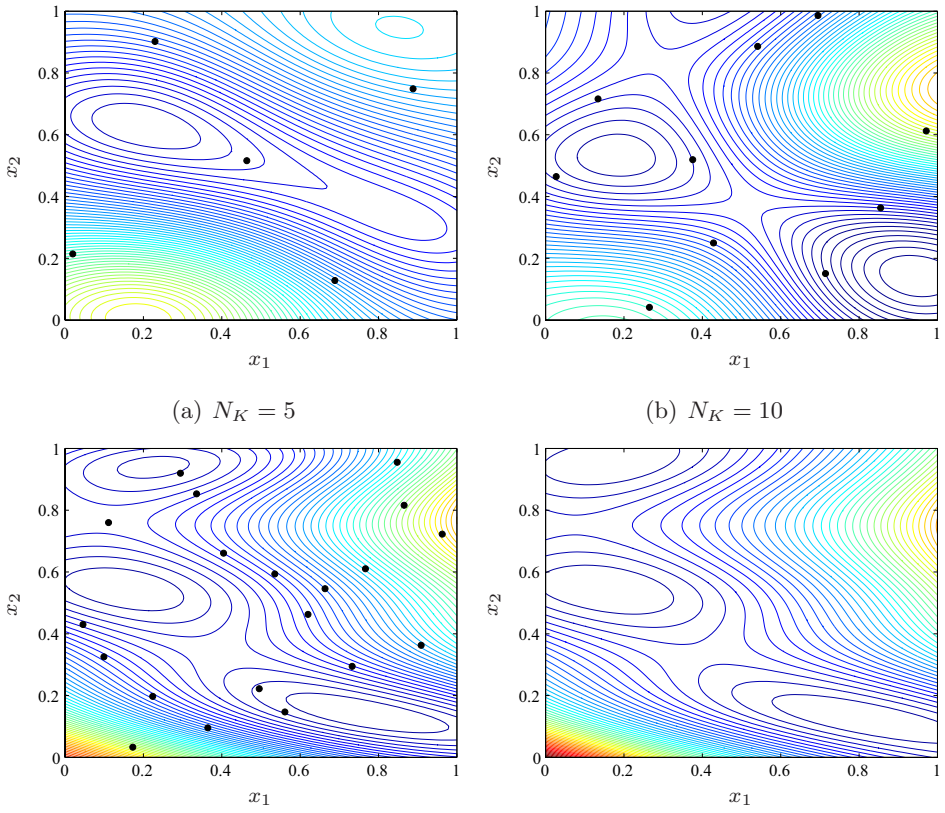
$$\tilde{\mathbf{R}} = \begin{pmatrix} \mathbf{R} & \mathbf{r} \\ \mathbf{r}^T & 1 \end{pmatrix}. \tag{4.12}$$

Substituting the expressions into Eq. (4.6), it is observed that only part of the augmented ln-likelihood function depends on the prediction $\hat{y}$, such that the function reduces to

$$\ln(L) \approx \frac{-\begin{pmatrix} \mathbf{Y} - \mathbf{1}\hat{\beta} \\ \hat{y} - \hat{\beta} \end{pmatrix}^T \begin{pmatrix} \mathbf{R} & \mathbf{r} \\ \mathbf{r}^T & 1 \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{Y} - \mathbf{1}\hat{\beta} \\ \hat{y} - \hat{\beta} \end{pmatrix}}{2\hat{\sigma}^2} \tag{4.13}$$

The augmented ln-likelihood function reduces to a quadratic function of $\hat{y}$ and is maximized by differentiating with respect to $\hat{y}$ and equating to zero,

$$\frac{\partial \ln(L)}{\partial \hat{y}} = \frac{-1}{\hat{\sigma}^2 \left(1 - \mathbf{r}^T \mathbf{R}^{-1} \mathbf{r}\right)} \left(\hat{y} - \hat{\beta}\right) + \frac{\mathbf{r}^T \mathbf{R}^{-1} \left(\mathbf{Y} - \mathbf{1}\hat{\beta}\right)}{\hat{\sigma}^2 \left(1 - \mathbf{r}^T \mathbf{R}^{-1} \mathbf{r}\right)} = 0 \tag{4.14}$$



(a) $N_K = 5$ (b) $N_K = 10$

(c) $N_K = 20$ (d) True Branin Function

Figure 4.5: Kriging approximations fitted to the *Branin* function based on N_K samples

The equation is rearranged such that the augmented ln-likelihood is maximized for the Kriging prediction at the location $\mathbf{x}$,

$$\hat{y}(\mathbf{x}) = \hat{\beta} + \mathbf{r}^T \mathbf{R}^{-1}(\mathbf{Y} - \mathbf{1}\hat{\beta}). \tag{4.15}$$

The *Branin* test function is used as an example of implementing a Kriging prediction based on initial datasets, as shown in Figure 4.5. The correlation parameter $\boldsymbol{\theta}$ is optimized based on maximum likelihood estimation using a pattern search method [Torczon, 1997]. The smoothness parameter is kept constant at $\mathbf{p} = 2$. Three independent Kriging models are constructed, based on LHS designs of $N_K = 5$, 10 and 20 respectively. Figure 4.5 features three different Kriging approximations, based on N_K highlighted samples. It is shown that as $N_K \rightarrow 20$ the Kriging approximation provides a very close *visual* representation of the true Branin function (shown in Figure 4.5(d)).

4.3.4 Model Validation

In the previous section, a Kriging approximation of the *Branin* function was demonstrated, and the results were visually presented through contour plots. The term *visually*

is emphasized here, since contour landscapes offer no real measure of the correlation of the dataset. In this section, methods to accurately test and validate the performance of Kriging models are considered. The inadequacy of a surrogate model can arise from saturation of the dataset, generating a noisy landscape, or insufficient sampling whereby the true relationship of the function landscape is not captured. Model-assessment strategies are therefore required to ensure the model is adequate for optimization [Keane and Nair, 2005]. Such methods are therefore useful in engineering design and assist the designer in making informed decisions about the subsequent optimization process.

Subset Validation

This method consists of reserving a separate subset of precise observations for validation of the Kriging model. To test the adequacy of the model, both the root mean squared error, or the correlation coefficient c^2 can be calculated [Forrester *et al.*, 2008]. Therefore, based on a subset of N_T observations, the performance of the model can be measured by:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=0}^{N_T} \left(y^i - \hat{y}^i\right)^2}{N_T}} \tag{4.16}$$

$$c = \frac{N_T \sum_{i=0}^{N_T} y^i \hat{y}^i - \sum_{i=0}^{N_T} y^i \sum_{i=0}^{N_T} \hat{y}^i}{\sqrt{\left[N_T \sum_{i=0}^{N_T} y^{(i)2} - \left(\sum_{i=0}^{N_T} y^i\right)^2\right] \left[N_T \sum_{i=0}^{N_T} \hat{y}^{(i)2} - \left(\sum_{i=0}^{N_T} \hat{y}^i\right)^2\right]}}. \tag{4.17}$$

The definition of the Root Mean Square Error (RMSE) means that it should ideally be as close to zero as possible, but dependent on the eventual application of the Kriging model it may not be necessarily required. For example, if the aim of constructing the surrogate is to visualize the entire design landscape then minimizing the RMSE is required. However if the surrogate is constructed to economically navigate the design landscape to identify the region of a global minimum, then a globally accurate Kriging model is not of great concern, and additional data points can be included using an update strategy.

The correlation coefficient is a very effective validation tool, and compares the shape of the landscapes rather than the values [Forrester *et al.*, 2008]. Figure 4.6 illustrates the relationship of the correlation coefficient to the sample dataset size N_K, for the *Branin* function. Results are generated based on a random subset of $N_T = 10$ samples. It is observed that there is some scatter in the results at low values of N_K, before the plot begins to plateau. It is clear that for $N_K \geq 18$ samples the correlation coefficient converges to one. At this point, generating a greater sample size will saturate the training dataset, and will not provide any higher prediction accuracy.

Although this method of validation is fairly conclusive, it requires a validation subset which is not always permissible for optimization problems, especially when there is a

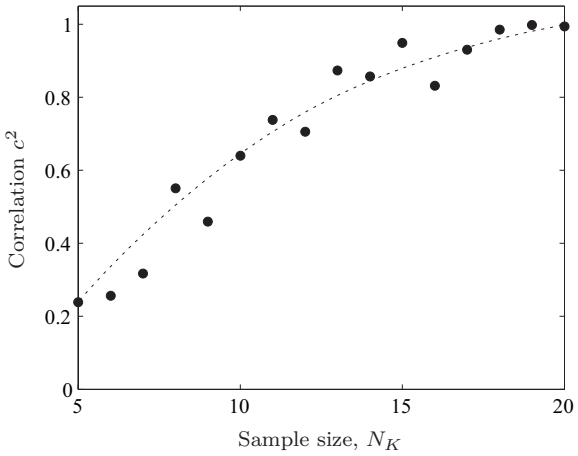


Figure 4.6: Relationship between correlation and sample size for *Branin* function

computational budget. The $N_T = 10$ samples used here for validation may be better utilized to update the Kriging model in promising areas of the design space. A method of cross-validation is hence considered, which does not require any additional evaluations whilst still providing a measure to the adequacy of the surrogate.

Cross Validation

The method of cross validation [Mitchell and Morris, 1992; Santers *et al.*, 2003] is a model assessment strategy which does not require the added expense of running additional evaluations. For practical optimization, the leave-one-out cross validation strategy is suggested by Santers *et al.* [2003]. In this approach, one sample point $\mathbf{x}_i$ is removed so that there are now $N_K - 1$ samples. The new Kriging model is rebuilt excluding the i-th sample. The difference between the prediction $\hat{y}(\mathbf{x}_i)$ and the precise evaluation $y(\mathbf{x}_i)$ is then computed. This process is repeated for all sample points $i = 1, \ldots, N_K$. The Cross Validation (CV) error is then calculated as,

$$\mathrm{CV} = \sqrt{\frac{\sum_{i=1}^{N_K} \left(y(\mathbf{x}_i) - \hat{y}(\mathbf{x}_i)\right)^2}{N_K}}. \tag{4.18}$$

In this expression, the root mean squared error of all predictions $\hat{y}(\mathbf{x}_i)$ are considered. The initial values for the parameter θ based on the model constructed with N_K samples is fixed during the validation process. Figure 4.7 demonstrates the application of the leave-one-out cross-validation scheme applied to Kriging models of varying N_K to fit the *Branin* function. It is fairly evident that the quality of the model increases as $N_K \to 20$, which is consistent with previous conclusions.

It is also possible to gain a visual understanding of the error spread. In Figure 4.8, the observed data is plotted against the Kriging predictions for four different sample plan

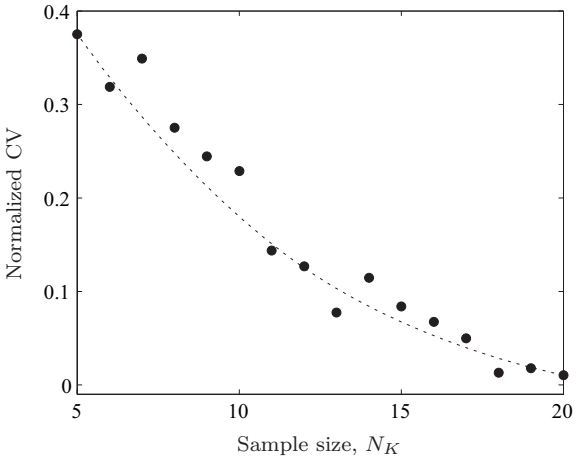
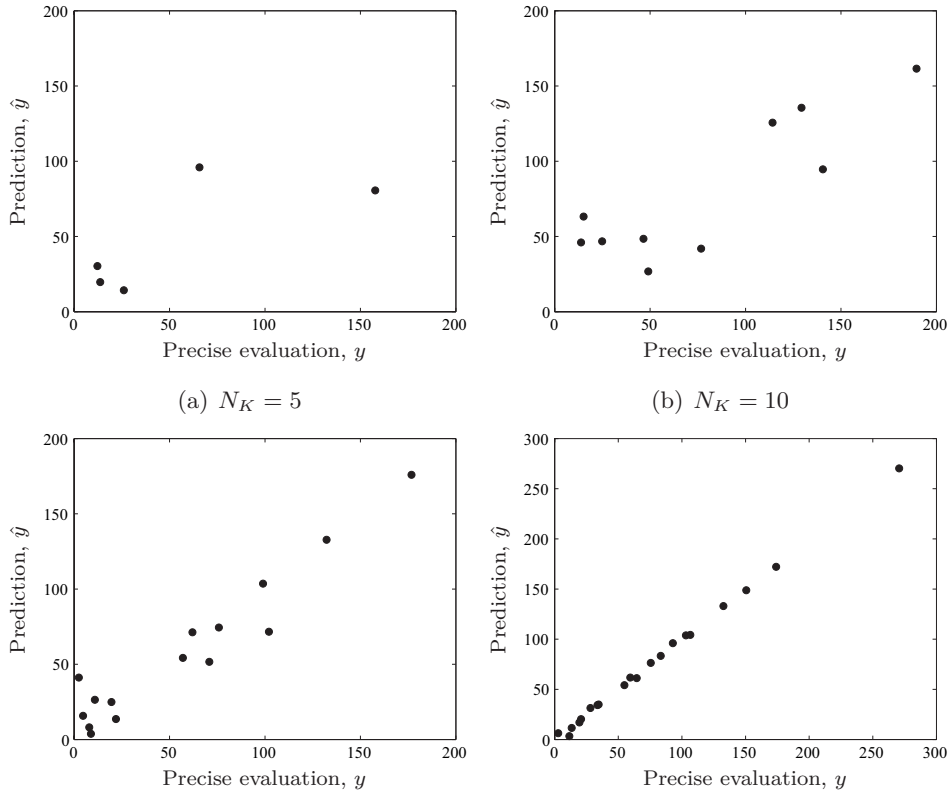


Figure 4.7: Relationship between CV error and sample size for *Branin* function

sizes. This is useful in order to obtain a qualitative measure of the model accuracy, and can provide insight into regions of the design space which are highly non-correlated. Non-correlated sample points can easily be identified, and the dataset can be conditioned to increase the global accuracy of the model. In the study by Emmerich *et al.* [2006] it is suggested that this validation process could also be used to adapt the value of θ during



(a) $N_K = 5$ (b) $N_K = 10$

(c) $N_K = 15$ (d) $N_K = 20$

Figure 4.8: Comparison between observed data and Kriging predictions for varying N_K

the training stage, replacing the maximum likelihood estimation, despite the increase in computing time. The authors further suggest the use of the median deviation rather than the average deviation, in order to increase robustness at outliers.

4.4 Visualization Techniques

Beyond their most obvious use of replicating an expensive function, surrogates can be used to filter or extract meaningful data from the function landscape. Visualization is used to either gain a functional insight between the input variables and the objective landscape, or perhaps visualize the compromise between identified trade-off solutions.

Optimization frameworks aim to identify the most optimal geometric shape for a given operating condition, however an analysis of the correlation between the design space and the objective space is not considered. Design space visualization identifies these relationships providing the designer insight into the optimization problem, and bridges the gap between the design space and the solution space both quantitatively and qualitatively. With the growing regard of surrogate modelling, the use of design space visualization and data mining techniques has progressively increased. However when confronted with a multi-dimensional design space and conflicting goals, identifying and isolating the case-specific information that suits the intended application can be challenging.

In this section, techniques for pre-optimization variable screening and post-optimization trade-off visualization are presented and discussed. It is demonstrated that extracting case-specific information from the visualization analyses, based on the preferred interests of the designer, is facilitated through the reference point. Variable influence and interaction, as well as important factors which drive the design are easily identified, which allow informed decisions to be made which may facilitate the optimization or decision-making processes.

4.4.1 Design Space Visualization

The most obvious form of visualization is concerned with graphically mapping the design space. This method allows for the key functional relationships between the inputs and outputs to be observed, to determine the modality or non-linearity of the design landscape. While this approach makes no statistical assumption on the design landscape, its main deficiency is its inability to analyze multi-dimensional data. Forrester *et al.* [2008] suggest that an intuitive way of gaining insight into a multi-dimensional landscape is to extract two-dimensional contours and strategically arrange them. The projections of these two dimensional slices stem from a baseline design, such that for a problem of n variables, $n-2$ variables are held at their respective baseline values, while two variables are visualized.

Variable	Description	Baseline	Min	Max
S_W	Wing area (ft^2)	174	150	200
W_{fw}	Fuel weight (lb)	252	220	300
A	Aspect ratio	7.52	6	10
Λ	quarter-chord sweep (deg)	0	-10	10
q	Cruise dynamic pressure (lb/ft^2)	34	16	45
λ	Taper ratio	0.672	0.5	1
t_c	Thickness to chord ratio	0.12	0.08	0.18
N_z	Load factor	3.8	2.5	6
W_{dg}	Gross weight (lb)	2000	1700	2500
W_p	Paint weight (lb/ft^2)	0.064	0.025	0.08

Table 4.1: Variable description and boundaries for the conceptual wing weight function

An analytical expression of the conceptual estimate of the weight of a light aircraft wing is considered as an example (extracted from Forrester *et al.* [2008], see Appendix A). Table 4.1 contains a description of the function variables, as well as a suggestion for a baseline design. Figure 4.9 shows a matrix of two-dimensional contour slices to visualize the non-linearity of the design landscape. It is immediately observed that the most influential variable (as expected) is the load factor (N_z) as it is involved interactions with other variables. Similarly the aspect ratio (A) is particularly influential, while variables such as the fuel weight (W_{fw}) and the paint weight (W_p) are not active at all. An insight into the fundamental aerodynamics of the problem is also obtained. Consider the two-

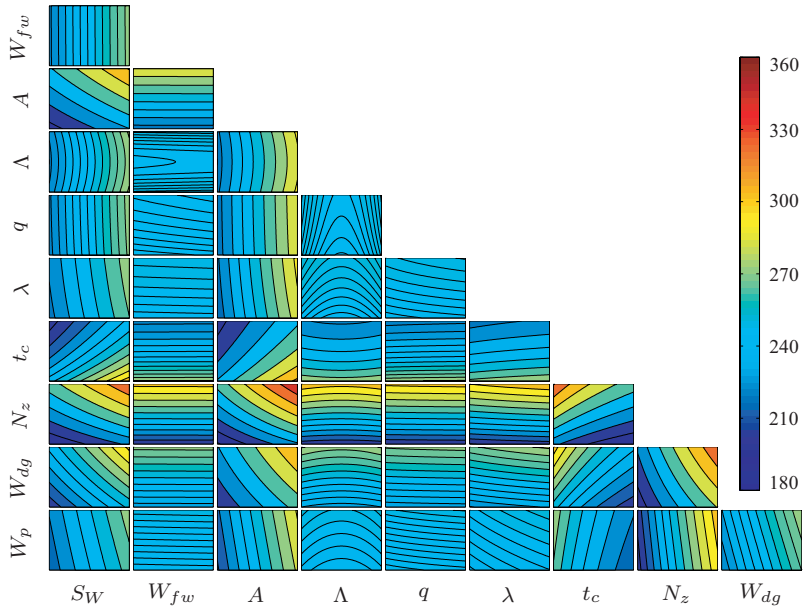


Figure 4.9: Matrix of two-dimensional contour plots (coloured by wing weight (lb))

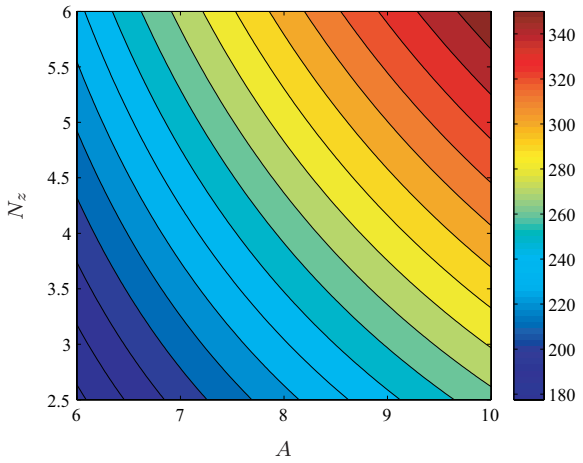


Figure 4.10: Two-dimensional contour of variables A and N_z (coloured by wing weight (lb))

dimensional contour slice of the variables A and N_z, which is shown in Figure 4.10. It is immediately observed that with increasing values of load factor and aspect ratio the wing weight approaches the maximum. This is the principal reason why highly manoeuvrable wings do not have a high aspect ratio, while glider aircraft have a large aspect ratio at the expense of limited manoeuvrability.

The example function is inexpensive, and the construction of a surrogate model is unnecessary. However for high-fidelity applications generating a multi-dimensional contour matrix requires an intractable computational effort, thus warranting the use of surrogate modelling, to provide a cheap-to-evaluate alternative at the expense of the prediction accuracy. While this method does provide an avenue to visualizing the design landscape, it is not immediately clear which variables are the most influential, and which variables are involved in non-linear interactions with other variables. The next section is concerned with screening methods which provide statistical measures to variable importance.

4.4.2 Variable Screening

The purpose of a screening study is to rank the design variables according to their elementary effect on the model $\hat{y}$. In many design problems, only few variables are primarily responsible for the variation in the model, whereas most of the other variables contribute linear or negligible effects [Trocine and Malone, 2000; Keane and Nair, 2005]. An estimate of the elementary effects of each variable identify the important factors which drive the design, and allow informed decisions to be made on the constriction of variable boundaries, or even the exclusion of a variable. A comparison of various screening methods used in engineering design is given by Trocine and Malone [2000] and Leary *et al.* [2002]. The authors discuss in detail the general framework for conducting variable screening studies.

There are several methods available for screening variables and each method is limited in one aspect or another. The designer is therefore required to make informed decisions about the appropriate method to apply, which may be based on the underlying structure of the model, the level of qualitative or quantitative information required, or the computational budget allocated for global sampling and visualization.

Variable screening has been considered in a number of aerodynamic design problems. For example, Jeong *et al.* [2005b] and Chiba *et al.* [2008] applied a screening method to multi-dimensional aerodynamic design problems where the most important variables corresponding to each objective were recognized. Khurana and Winarto [2010] conducted a screening analysis for airfoil shape design, where it was concluded that one of the variables could be safely omitted from the subsequent optimization study. A recurring strategy in these studies is to combine the sensitivity studies with visualization techniques to concisely present quantitative information about the design space. Presented in this section are two methods which are useful for identifying the sensitivity and elementary effects of variables.

Analysis of Variance

Perhaps the most intuitive method of determining the sensitivity of each variable, is through an analysis of variance [Sobol, 1993; Jones, 1998]. This method is particularly useful in analyzing design problems of many dimensions, where a complex problem may be simplified by identifying non-active variables [see, for example Jeong *et al.*, 2005b; Chiba *et al.*, 2008]. The total variance of the model $\hat{y}$ is decomposed into the contributions from individual variables and from variable interactions. A measure to the influence of each design variable on the model is therefore obtained.

Decomposition of the model variance is simply achieved by integrating variables out of the model $\hat{y}$. Consider the model $\hat{y}(\mathbf{x})$ where the variables $\mathbf{x}$ lies in the unit interval $[0,1]^n$. The total mean $\hat{\mu}$ and variance $\hat{\sigma^2}$ can be calculated as follows:

$$\hat{\mu} = \int_0^1 \dots \int_0^1 \hat{y}(x_1 \dots x_n) dx_1 \dots dx_n, \tag{4.19}$$

$$\hat{\sigma^2} = \int_0^1 \dots \int_0^1 \left[\hat{y}(x_1 \dots x_n) - \hat{\mu}\right]^2 dx_1 \dots dx_n. \tag{4.20}$$

To assess the main effect of variable x_i, all other variables are integrated out of the $\hat{y}$. This results in a function solely of x_i, which is denoted $\hat{\mu}_i(x_i)$:

$$\hat{\mu}_i(x_i) = \int_0^1 \dots \int_0^1 \hat{y}(x_1 \dots x_n) dx_1 \dots dx_{i-1} dx_{i+1} \dots dx_n - \hat{\mu}. \tag{4.21}$$

Similarly, the two-way interaction effect between the variables x_i and x_j is denoted as:

$$\hat{\mu}_{ij}(x_i, x_j) = \int_0^1 \dots \int_0^1 \hat{y}(x_1 \dots x_n) dx_1 \dots dx_{i-1} dx_{i+1} \dots dx_{j-1} dx_{j+1} \dots dx_n - \hat{\mu}_i(x_i) - \hat{\mu}_j(x_j) - \hat{\mu}. \quad (4.22)$$

The variance due to the design variable x_i is quantified as

$$\hat{\sigma}_i^2 = \int_0^1 \left[\hat{\mu}_i(x_i)\right]^2 dx_i, \quad (4.23)$$

thus the proportionate variance of x_i to the total variance of the model $\hat{\sigma^2}$ is

$$\frac{\int_0^1 \left[\hat{\mu}_i(x_i)\right]^2 dx_i}{\int_0^1 \dots \int_0^1 \left[\hat{y}(x_1 \dots x_n) - \hat{\mu}\right]^2 dx_1 \dots dx_n}. \quad (4.24)$$

For certain polynomial models the above integrals can be performed analytically [Keane and Nair, 2005]. For Kriging models, sampling plans can be used to numerically approximate the integrals. The subsequent method does not require the approximation of integrals and is applicable to any deterministic function. In the opinion of the author, this is a more concise method of measuring the level of activity of each variable.

Morris Screening Method

A popular method for designing preliminary experiments for design space visualization is the screening method developed by Morris [1991] and discussed in detail also in Forrester *et al.* [2008]. If the model $\hat{y}$ is differentiable over the design space $\mathcal{S}$ with respect to x_i, then the effect of $x_i \in \mathcal{S}$ can be established as:

- Negligible, if $(\partial\hat{y}/\partial x_i) = 0$,
- Linear, if $(\partial\hat{y}/\partial x_i) = \text{constant} \neq 0$,
- Nonlinear, if $(\partial\hat{y}/\partial x_i) = f(x_i) \neq \text{constant}$,
- Nonlinear and interacting with $x_j, \dots$, if $(\partial\hat{y}/\partial x_i) = f(x_i, x_j, \dots) \neq \text{constant}$.

For an engineering function, there is no real measure for computing $(\partial\hat{y}/\partial x_i)$ across the entire design space $\mathcal{S}$. In plain terminology, the Morris algorithm measures the sensitivity of the i-th variable to the model $\hat{y}$ through finite differencing. The design space $\mathcal{S}$ is restricted to an n-dimensional, p-level full factorial grid, where the normalized variable,

$$x_i \in \{0, 1/(p-1), 2/(p-1), \dots, 1\}, \text{ for } i = 1, \dots, n. \quad (4.25)$$

For a given baseline design $\mathbf{x}$, $d_i(\mathbf{x})$ denotes the elementary effect of x_i, where

$$d_i(\mathbf{x}) = \frac{\hat{y}(x_1, x_2, \dots, x_{i-1}, x_i + \Delta, x_{i+1}, \dots, x_n) - \hat{y}(\mathbf{x})}{\Delta}, \quad (4.26)$$

Variable	μ_i	σ_i	Influential	Interactive
S_W	12.0373	60.395	✓	✓
W_{fw}	0.0505	0.2929		
A	-10.6727	77.5943	✓	✓
Λ	1.6341	8.0751		
q	-0.5714	1.566		
λ	-0.7615	7.9458		
t_c	2.9585	77.0339		✓
N_z	46.3847	113.2279	✓	✓
W_{dg}	4.2889	50.8157		✓
W_p	1.1917	9.7916		

Table 4.2: Determining wing weight variable effects using the Morris method

where $\Delta = \xi/(p-1)$, $\xi \in \mathrm{IN}^*$ is the elementary effect step length factor, and $\mathbf{x} \in \mathcal{S}$ such that $x_i \leq 1 - \Delta$. Morris' method estimates the distribution of elementary effects of x_i over $\mathcal{S}$. If the variable x_i has a large central tendency μ_i, then this suggests an important influence on $\hat{y}$. A large measure of spread σ_i suggests that either x_i is involved in interactions with other variables or is nonlinear with $\hat{y}$. Essentially, the sample mean and standard deviation of a set of $d_i(\mathbf{x}) \in \mathcal{S}$ values are calculated. Clearly, it is desirable to maximize the spread *and* number of elementary effects to provide a more thorough representation of the design space, without compromising the global accuracy of the model. A method to generate a sampling plan to estimate r elementary effects is given by Morris [1991].

Referring back to the conceptual wing weight function, it is possible to determine the elementary effect of each variable using Morris' screening method. Table 4.2 consolidates the results of the screening analysis and based on the values of central tendency and variance, determines whether the variables are influential and/or interacting.

As discussed by Saltelli *et al.* [2001], an extension to the original Morris method introduces a ranking measure μ_i^*, which is calculated from the mean of the distribution of the absolute values of the elementary effects, such that

$$\mu_i^* = \overline{|d_i(\mathbf{x})|}. \tag{4.27}$$

The proposed methodology addresses the misrepresentation of the magnitude of sensitivity, as a result of opposite signs of the elementary effects for a non-monotonic model. To rank variables in order of influence or importance, the quantity μ^* may also be used. Figure 4.11 illustrates the results of the revised screening method, where the proportion of variance of μ^* for each variable (and the corresponding ranking) is observed. It is immediately

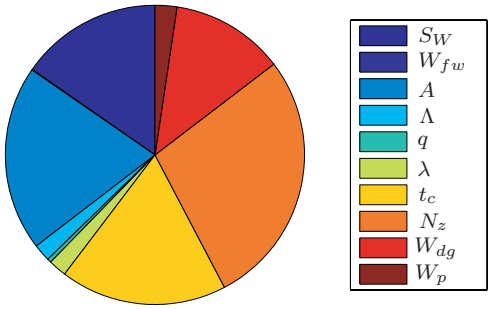


Figure 4.11: Pie-chart of proportional variance for conceptual wing weight function

observed that the most influential variable is the load factor (N_z), while other variables of significant influence are the wing area (S_W), the aspect ratio (A), the thickness to chord ratio (t_c) and the gross weight (W_{dg}). Of the least influential variables it is concluded that the dynamic pressure (q) and the fuel weight (W_{fw}) have a negligible effect, and could be safely omitted from the subsequent optimization analysis - a conclusion which is otherwise not possible to reach solely from Figure 4.9. Thus the design space is reduced to eight dimensions which will accelerate the search effort and alleviate the computational effort of constructing and updating the Kriging model.

4.4.3 Self-organizing Maps

To ideally represent the identified trade-offs and the compromise between objectives, a data mining technique known as Self-Organizing Maps (SOM) is introduced. Clustering SOM techniques are based on the analogy of a neural net to organize and visualize large volumes of data to facilitate its interpretation. The methodology is based on a technique of unsupervised artificial neural network to model the nonlinear projection from high-dimensional inputs to the self-organization of a low-dimensional array of neurons. For a comprehensive discussion on SOM variants, the reader is referred to the excellent texts of Fausett [1994] and Kohonen [1995]. In this section a brief introduction to SOM is provided, extracted from Jeong *et al.* [2005b] and Khurana and Winarto [2010].

SOM techniques are primarily applied for pattern recognition and for clustering and visualizing large volumes of data [Fausett, 1994]. An example of a SOM structure is presented in Figure 4.12, where a 4×4 network of neurons are directly connected to the input nodes, but not directly inter-connected [see Khurana and Winarto, 2010]. The neurons are positioned two-dimensionally for ease of visualization. In the projection algorithm of SOM, the connecting weights between the input vector and the array of neurons are adjusted to represent features of the high-dimensional data on the low-dimensional map.

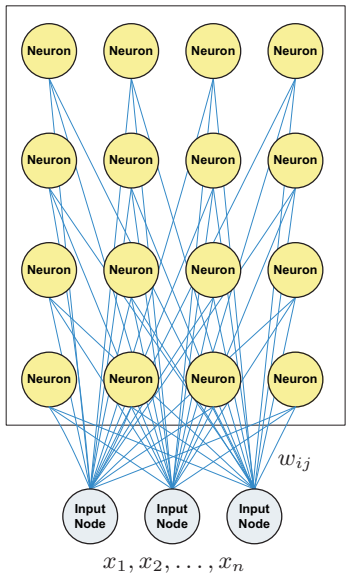


Figure 4.12: Simple representation of a SOM structure [see Khurana and Winarto, 2010]

A neuron has the weight vector $\mathbf{w}_i = [w_{i1}, w_{i2}, \ldots, w_{in}]$ $\forall i = 1, \ldots, N$ where n is the number of input variables and N is the number of neurons. Each neuron is connected to its adjacent neurons by a neighborhood relation forming either a one or two-dimensional rectangular or hexagonal topology. The learning algorithm will attempt to maximize the correlation between neighbouring neurons by finding the best-matching unit ($\mathbf{w}_c$):

$$\|\mathbf{x} - \mathbf{w}_c\| = \min \|\mathbf{x} - \mathbf{w}_i\| \quad \forall i = 1, \ldots, N \tag{4.28}$$

The best-matching units are determined sequentially, and for each neuron the weight adjustments are performed for the respective local neighbours to organize the topological mapping. With repeated iterations of this learning algorithm the adjustments lead to a global correlation [see Jeong *et al.*, 2005b].

In this research, SOM are generated using the software Viscovery SOMine 5.2. SOMine constructs SOM charts via an advanced variant of unsupervised neural networks, known as the Kohonen batch [Kohonen, 1995]. Similar to the sequential learning algorithm, the Kohonen batch algorithm searches for the best-matching unit c_i for all input data $\mathbf{x}_i$, and adjusts the weight vector $\mathbf{w}_j$ locally surrounding the best-matching unit. The Batch SOM algorithm can be formulated as follows:

$$c_i = \min_j \|\mathbf{x}_i - \mathbf{w}_j\| \tag{4.29}$$

$$\mathbf{w}*_j = \sum_i h_{j_{c_i}} \mathbf{x}_i / \sum_i h_{j_{c_i}} \tag{4.30}$$

where $\mathbf{w}*_j$ is the adjusted weight vector. The neighborhood relationship between the

neuron j and the best-matching unit c_i is defined by the following function:

$$h_{j_{c_i}} = \exp\left(-\frac{d^2_{j_{c_i}}}{r_t^2}\right) \tag{4.31}$$

where $d_{j_{c_i}}$ is the Euclidean distance from the neuron j to the best-matching unit c_i on the map, and r_t denotes the neighborhood radius, which is gradually decreased with increasing number of iterations t.

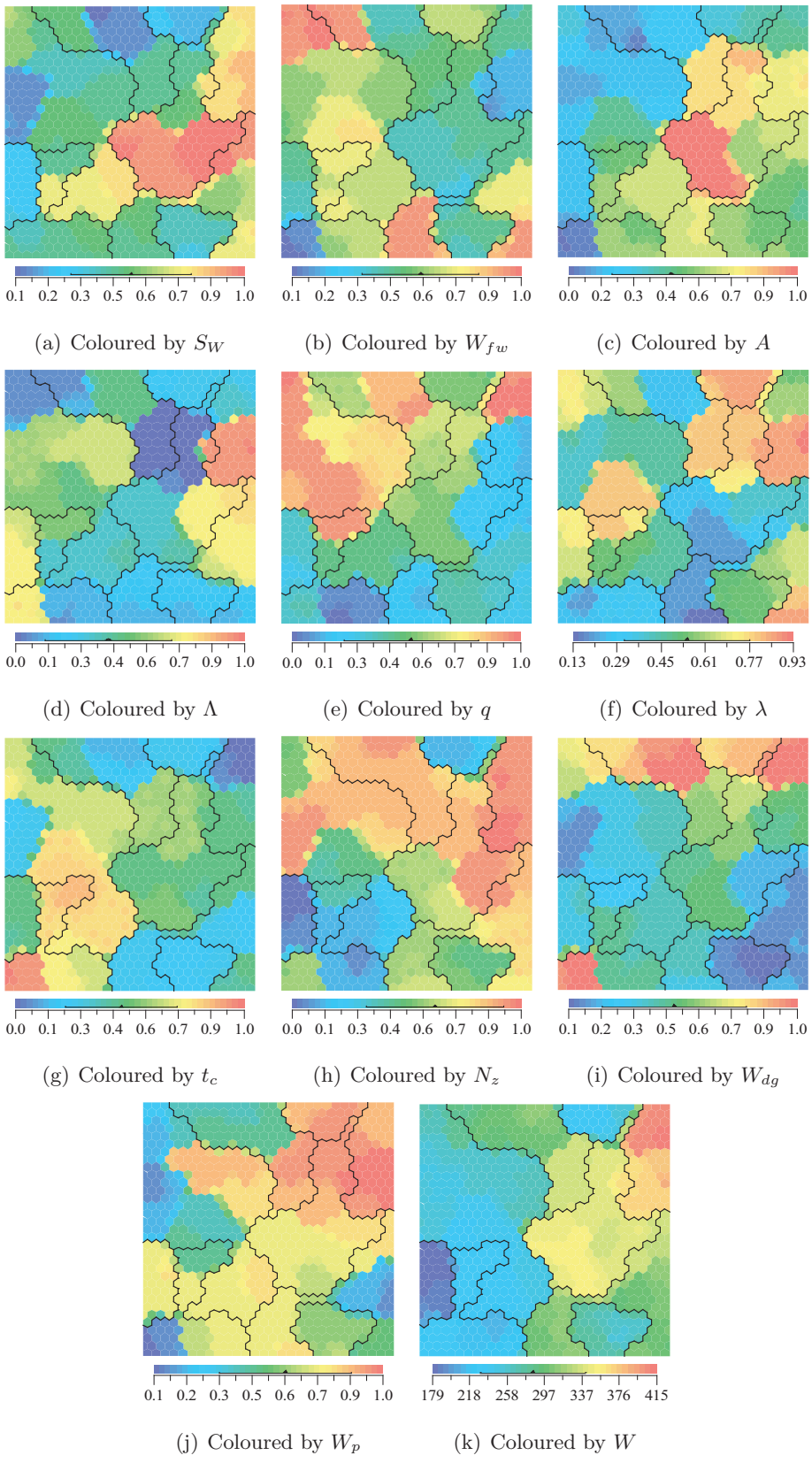
The Kohonen batch algorithm performs a collected adjustment of the weight vectors apart from individually or sequentially adjusting the weight vector. SOMine creates two-dimensional SOM charts arranged in a hexagonal topology. Based on the presentation of the input data, the resulting chart will reflect the characteristics of the data distribution. Once the input data is projected and arranged on the two-dimensional grid, the SOM can be used for qualitative visualization. A hierarchical algorithm is used to cluster the data, based on priorities specified by the user. SOM are organized into a user-defined number of clusters, to facilitate the interpretation of the charts.

To illustrate the function of SOMine, refer to the conceptual wing weight function introduced earlier. A stratified sample of the multi-dimensional design space is first obtained, and is organized in no particular fashion. The corresponding outputs are arranged by increasing values. Output values are assigned a user-defined integer to denote the respective data cluster and facilitate the clustering process. The sample data is therefore arranged as:

$$\begin{bmatrix} S_W & W_{fw} & A & \cdots & W_p \\ 0.7653 & 0.5355 & 0.1691 & \cdots & 0.2333 \\ 0.0522 & 0.7701 & 0.5865 & \cdots & 0.9973 \\ 0.9912 & 0.0708 & 0.4827 & \cdots & 0.4300 \\ 0.1347 & 0.1912 & 0.6906 & \cdots & 0.0874 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0.2519 & 0.4370 & 0.9164 & \cdots & 0.5249 \end{bmatrix} = \begin{bmatrix} W \\ 184.7479 \\ 191.8359 \\ 198.7431 \\ 205.4847 \\ \vdots \\ 358.1698 \end{bmatrix} = \begin{bmatrix} Cluster \\ 1 \\ 1 \\ 1 \\ 1 \\ \vdots \\ 10 \end{bmatrix} \tag{4.32}$$

The data has been arranged into 10 clusters, based on increasing values of the output W[2]. Figure 4.13 features the results of the SOM analysis. The SOM charts are topologically aligned via the data clusters. To read the charts it is required to sequentially monitor each data cluster to arrive at any meaningful conclusion. For example, the cluster of minimum

[2]Reference has been made to inputs and outputs for ease of interpretation, however SOMine treats each column of data as an input.



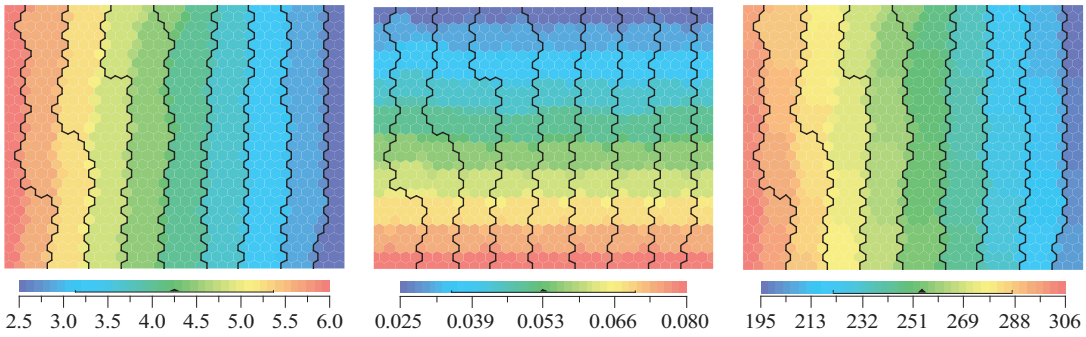
(a) Coloured by S_W (b) Coloured by W_{fw} (c) Coloured by A

(d) Coloured by Λ (e) Coloured by q (f) Coloured by λ

(g) Coloured by t_c (h) Coloured by N_z (i) Coloured by W_{dg}

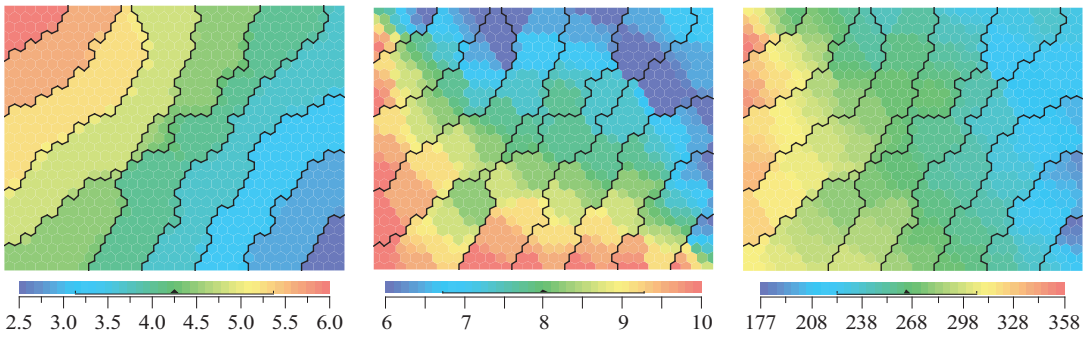
(j) Coloured by W_p (k) Coloured by W

Figure 4.13: Visualizing a stratified sample of the conceptual wing weight design function via SOM representation



(a) Coloured by N_z (b) Coloured by W_p (c) Coloured by W

Figure 4.14: Visualizing the compromise between key variables via SOM representation



(a) Coloured by N_z (b) Coloured by A (c) Coloured by W

Figure 4.15: Visualizing the compromise between key variables via SOM representation

values of wing weight W (the bottom left section of Figure 4.13(k) corresponds to low values of wing area S_W, low values of aspect ratio A, high values of dihedral Λ, etc.

The data however is ill-represented and apart from making the most obvious conclusions, the maps are very difficult to interpret to extract any meaningful insight. For example, it was determined in earlier studies that the paint weight W_p is essentially inactive and not involved in any interactions with other variables. From Figure 4.13(j) however, it is not possible to arrive to the same conclusion, and the observed scatter in the chart could either represent noise or just as likely it could represent high non-linearity. The input data should therefore be systematically arranged in order to arrive at meaningful conclusions. Figure 4.14 provides an example of systematically arranged data. In this case, data is clustered to measure the interaction between variables N_z and W_p on the wing weight W. It is now clearly evident that the paint weight W_p is essentially inactive, since varying the values of W_p together with increasing values N_z produces a negligible change to the wing weight W. Similar conclusions can be derived for other variables, for example Figure 4.15 features the influence and interaction between the variables N_z and A for the

wing weight W. It is clearly shown that the wing weight W is highly sensitive to the interactions between both variables. Obviously the analyses shown in Figures 4.14 and 4.15 are useless since similar conclusions may be obtained using design space contours. It has however illustrated the necessity to strategically arrange the data input which facilitates the interpretation of the SOM charts.

4.5 Summary

This chapter provides an introduction to surrogate modelling and its use and integration within an evolutionary optimization framework. Surrogate models are used in lieu of the expensive function to efficiently navigate the optimization landscape. However, to construct a globally accurate surrogate of the entire design landscape is improbable and computaitonally expensive. It is more advantageous to generate a coarse represention of the design space using a stratified sample dataset, and locally updating the surrogate as promising areas of the design space are identified by the optimizer. A review of some popular methodologies to construct sample datasets and integrate a surrogate within the optimization architecture is provided. Focus is thereafter given to introducing the Kriging method, which is of particular importance in this research. Procedures to train, construct and validate a Kriging model are provided and demonstrated. The chapter concludes with yet another beneficial use of surrogates, namely for visualization and data mining. Variable screening and sensitivity methods to visualize the design space and variable activity are described. While such methods do not explicitly require the construction of a surrogate, the aid of a surrogate model is highly advanatageous. A data mining technique known as self-organizing maps (based on an unsupervised neural network) is further introduced, whereby multi-dimensional data is effectively visualized using an array of two-dimensional maps. Common examples of the use of the aforementioned visualization methods are provided, providing some insight to their applicablity for this research.

Chapter 5

Novel Preference-based MOPSO Algorithm

Evolutionary computing is a recurring theme for multi-objective aerodynamic design frameworks. While most population-based algorithms mimic the competitive theory of survival of the fittest, the concept of maintaining a population of *collaborating* individuals is increasingly becoming popular. In this research, the multi-objective search effort is coordinated via a particle swarm optimization algorithm. A variant to the canonical swarm algorithm is developed, which draws on the domain knowledge of the designer to obtain solutions of interest. The swarm is guided by a reference point, which is an intuitive means of articulating the designer preferences and can ideally be based on an existing or target design. This chapter provides a comprehensive discussion on the proposed algorithm and a visual demonstration of its operation. A series of test function suites are investigated, which highlight the viability of the proposed framework for the intended domain of application.

5.1 Incorporating Designer Preferences

The fundamental PSO architecture was provided in Chapter 3. The dynamics of the swarm were described, and the methodology to balance exploration-exploitation characteristics, as well as overcoming boundary violation was documented. An introduction to the extension of PSO to handle or multiple objectives (or MOPSO) was subsequently provided in Section 3.5.3. This novel algorithm aims to integrate the recent developments in MOPSO to the preference-based architecture. The principal argument of this research is that for most design applications, to explore the entire Pareto front is often unnecessary and the computational burden can be alleviated by considering the immediate interests of the designer. In Section 3.5.4, a discussion on the benefits of preference-based optimization was

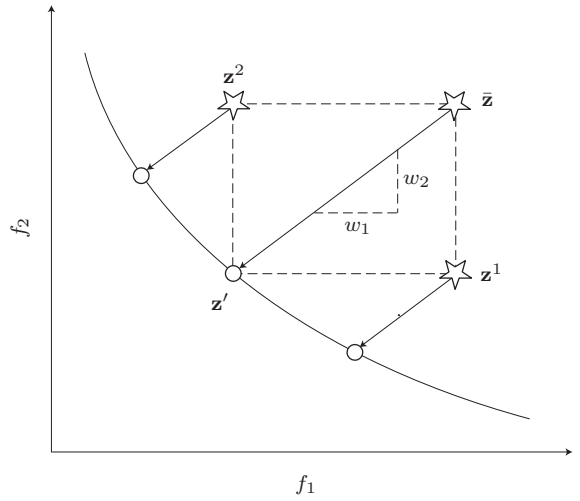


Figure 5.1: The classical reference point approach

provided. Drawing on these concepts, a preference-based algorithm is proposed, where a designer-driven distance metric is used to scalar quantify the success of a solution. The swarm is guided by this information to confine its search focus exclusively on the preferred region of the Pareto front as dictated by the preferences of the designer. Introducing the preferred region provides the designer flexibility to explore other interesting alternatives. This hybrid methodology is advantageous to navigate high-dimensional and multi-modal landscapes, which are typical of aerodynamic design problems. Furthermore, inherently considering the preferences of the designer provides a feasible means of quantifying the practicality of a design.

5.1.1 The Reference Point Method

In order to introduce the preference metric utilized in this algorithm, it is beneficial to gain an understanding of the preferred compromise or the *reference point*. This classical method first introduced by Wierzbicki [1980], directly uses the reference point to identify preferred designs. The reference point is used to construct a single-objective optimization which is minimized for $\mathbf{x} \in \mathcal{S}$,

$$\text{minimize} \quad \max_{i=1:m} \left\{ w_i \cdot \left(f_i \left(\mathbf{x} \right) - \bar{z_i} \right) \right\}, \tag{5.1}$$

where $\bar{\mathbf{z}} = \{\bar{z_1}, \ldots, \bar{z_m}\}$ is the reference point of m reference values and $\mathbf{w} = \{w_1, \ldots, w_m\}$ is a vector of weights indicating the priority of each objective. The classical reference point method is illustrated in Figure 5.1 for a bi-objective minimization problem. For a given reference point $\bar{\mathbf{z}}$, the optimal solution is the target solution of the reference point method $\mathbf{z}'$. Wierzbicki [1980] further suggested a procedure whereby the optimal solution $\mathbf{z}'$ is used to identify m new reference points as shown in Figure 5.1. Subsequently, these

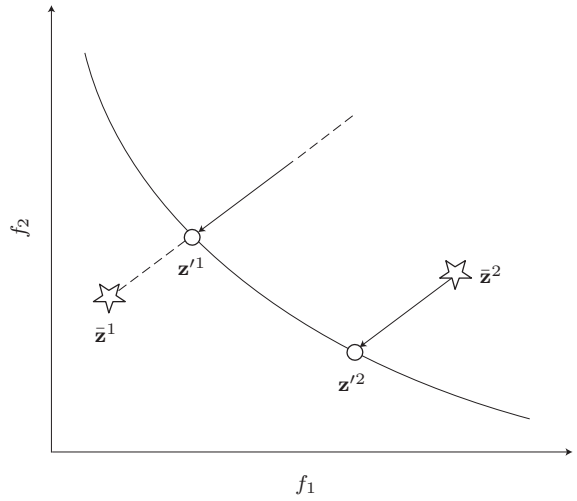


Figure 5.2: Illustration of the search direction governed by the reference point

identified reference points may be used to derive additional solution points to provide a solution spread. The region surrounding the reference point $\bar{\mathbf{z}}$ is known as the preferred region, and provides a measure of the designer's confidence in the reference point selection.

5.1.2 The Reference Point Distance Metric

The reference point method has been integrated into MOO algorithms, notably by Deb and Sundar [2006] and Wickramasinghe and Li [2008; 2009]. These studies highlight the benefits of incorporating preference information via the reference point in terms of convergence. Guided by the information provided by the reference point, the swarm can simultaneously identify multiple solutions in the preferred region. This provides the designer flexibility to explore several *preferred* designs, while alleviating the computational burden of identifying the entire Pareto front. A reference point distance metric following the work of Wickramasinghe and Li [2008] is proposed. This metric provides an intuitive criterion to select global leaders, and assists the swarm to identify only solutions of interest to the designer. The distance of a particle $\mathbf{x}$ to the reference point $\bar{\mathbf{z}}$ is defined as:

$$d_z(\mathbf{x}) = \max_{i=1:m} \left\{ \left(f_i\left(\mathbf{x}\right) - \bar{z}_i \right) \right\}. \tag{5.2}$$

The weight vector $\mathbf{w}$ is omitted to give equal priority to each objective. A arbitrary solution $\mathbf{a}$ is therefore *preferred* to the arbitrary solution $\mathbf{b}$ if $d_z(\mathbf{a}) < d_z(\mathbf{b})$. This condition is an extension of the condition $\mathbf{f}(\mathbf{a}) \prec \mathbf{f}(\mathbf{b})$, therefore the distance metric may in fact substitute the dominance criteria entirely [see Wickramasinghe and Li, 2009]. Using this distance metric, the swarm is guided to preferred regions of the Pareto front. Figure 5.2 illustrates the search directions of the algorithm when guided by the j-th reference point $\bar{\mathbf{z}}^j$, and the corresponding preferred design $\mathbf{z}'^j$ as a result of minimizing the distance metric.

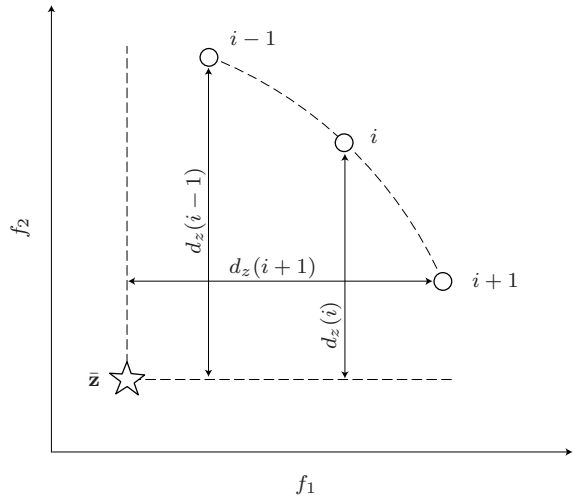
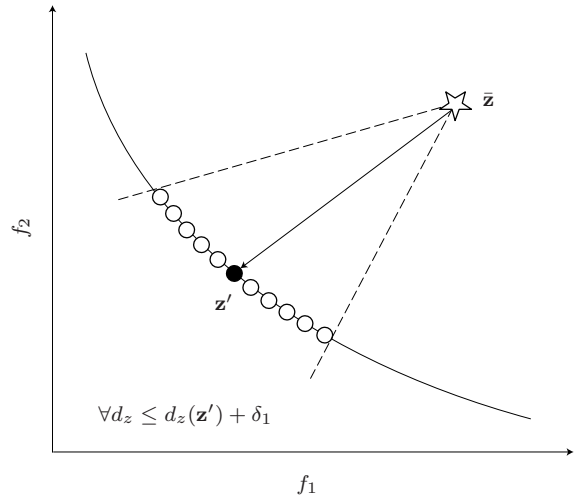


Figure 5.3: Identifying the preferred solution from solutions with equal Euclidean distance

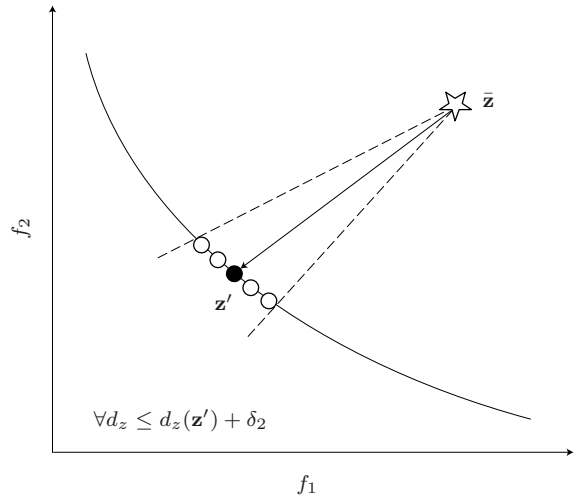
The distinguishing feature of the reference point distance metric over the mathematical Euclidean distance is that solutions do not converge to the reference point, but on the preferred region of the Pareto front as dictated by the search direction (which is dependent only on the relative location of the reference point to the Pareto front). This is illustrated in Figure 5.3. All solutions are non-dominated and lie on the circular arc surrounding the reference point $\bar{\mathbf{z}}$ and thus the Euclidean distance to the reference point is equal. However since solution i has the smallest maximum translational distance $d_z(i)$ to the reference point compared to any other solution, it is considered preferred. The definition of the reference point distance also therefore permits solutions to have negative values. In the case that the distance of the preferred solution $d_z(\mathbf{z}') < 0$ then it can simply be considered that the reference point is dominated or $\mathbf{z}' \prec \bar{\mathbf{z}}$. Since the designer generally has no prior knowledge of the topology and location of the Pareto front, a reference point may be ideally placed in any feasible or infeasible region, as shown in Figure 5.2. It is therefore the consensus that the reference point draws on the experience of the designer to express the preferred compromise, rather than specific target values or goals. Similarly, the reference point distance metric ranks or assesses the success of a particle as one single scalar, instead of an array of objective values.

5.1.3 Defining the Preferred Region

As demonstrated in Figure 5.2, if there is no control over the solution spread, the swarm will explore the preferred search direction and converge to the single solution $\mathbf{z}'$ as dictated by the reference point $\bar{\mathbf{z}}$. The advantage of maintaining a population of particles provides the designer with the possibility to explore a range of interesting alternatives within a preferred region of the Pareto front. The aim is therefore to identify a set of solutions



(a) $\delta_1 = 0.01$



(b) $\delta_2 = 0.001$

Figure 5.4: Definition of the preferred region via the parameter δ

surrounding the intersection point $\mathbf{z}'$. A threshold parameter $\delta > 0$ is defined, such that a solution $\mathbf{x}$ is within the preferred region if the following conditional statement is true:

$$d_z(\mathbf{x}) \leq d_z(\mathbf{z}') + \delta \tag{5.3}$$

Figure 5.4 illustrates the definition of the preferred region. The extent of the solution spread is proportional to δ and evidently as $\delta \to 0$, the designer is interested in determining only the most preferred solution $\mathbf{z}'$. Conversely, as $\delta \to \infty$, the designer is interested in determining all solutions along the Pareto front, and thus the influence of the reference point location diminishes. The function of the solution spread implies that if the designer were supremely confident that the reference point is a viable reflection of the preferred compromise, then a smaller solution spread may be used to increase search efficiency.

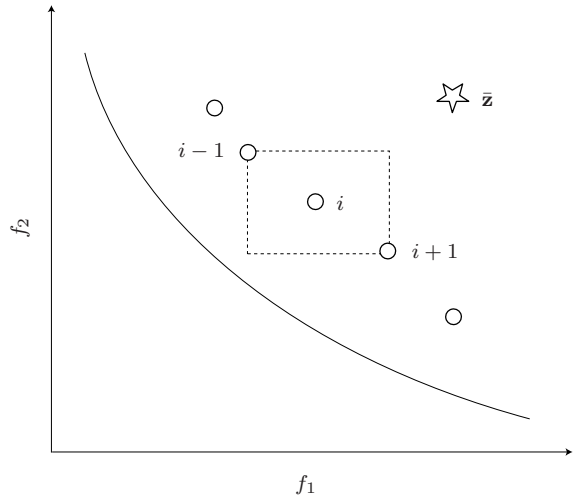


Figure 5.5: Crowding distance computation of solution point i

Otherwise if there is doubt in selection of the reference point, then a larger spread provides the flexibility to explore other interesting alternatives during post-optimization.

5.1.4 Solution Clustering

The introduction of δ encourages a broader search region, and control measures must be implemented to ensure a uniform solution spread. Niching methods are extensively studied and used to maintain population diversity in EMO algorithms [Deb, 2001]. A commonly used method is the sharing function introduced by Goldberg and Richardson [1987]. In this method, the niche count of a particle is defined as the number of other particles within a distance σ_{share}. The undesirable feature is that the value of σ_{share} must be specified, and the performance of the metric is highly dependent on this parameter value. Fonseca and Fleming [1993] extended this approach by developing a dynamic update of σ_{share}, based on the upper and lower bound values for each objective at each iteration.

In this algorithm the crowding distance method is utilized, which is free of choosing such a parameter. This method was first introduced by Deb *et al.* [2002]. The crowding distance metric has also been applied to MOPSO algorithms, notably in the work of Li [2003]; Raquel and Naval [2005]; Reyez-Sierra and Coello Coello [2005]. The crowding distance is calculated by first sorting the set of solutions in ascending objective function values. The crowding distance of a solution i is the size of the largest cuboid bounded by the solutions $i-1$ and $i+1$ without enclosing any other solution, as illustrated in Figure 5.5. The final value is obtained by adding the individual distances for each objective:

$$d_c(i) = \sum_{j=1}^{m} \frac{f_{(i+1,j)} - f_{(i-1,j)}}{(f_{\max,j}) - (f_{\min,j})} \tag{5.4}$$

It is shown that the crowding distance is essentially the Euclidean distance from its two neighbouring solutions. Boundary solutions are given an infinite crowding distance value to increase selection pressure.

5.2 Identifying Preferred Solutions

User-preferences alleviate the computational burden of identifying the entire Pareto front. While the concept of the reference point is fairly intuitive, ensuring the swarm is guided by this information to identify preferred solutions is more ambiguous. This section discusses the synchronization of the searching prowess of the MOPSO algorithm with the guidance of the reference point. Leader selection schemes incorporate the reference point information to guide the swarm along the preferred search path. Additional features incorporated, such as a mutation operator to prevent convergence to local fronts, are also highlighted.

5.2.1 Global Best Selection Scheme

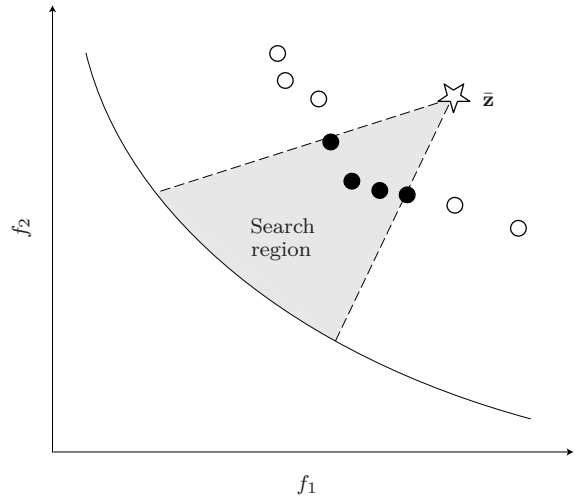
The methodology for updating and storing non-dominated solutions is primarily influenced by the MOPSO variants of Reyez-Sierra and Coello Coello [2005], and Raquel and Naval [2005]. A secondary population of particles in the form of an elitist archive (denoted by $\mathbf{Q}$) is maintained, such that

$$\mathbf{Q} = \{\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_K\}^T \tag{5.5}$$

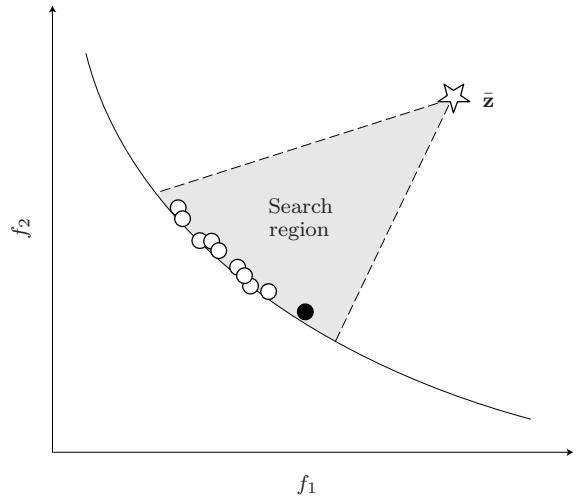
where $\mathbf{q}$ refers to a non-dominated solution, and K is the number of archive members at time, t. After the population update $\mathbf{x}(t)$ to $\mathbf{x}(t+1)$, the non-dominated solutions identified by the particles are appended to the archive. The archive $\mathbf{Q}(t)$ with the new additions is then updated to $\mathbf{Q}(t+1)$ using a non-dominated sorting procedure where all members pertaining to local inferior fronts are omitted. The archive serves as a mutually accessible memory bank for the particles of the swarm. Each member is a potential candidate for global leadership of the particles during the subsequent velocity update.

Defining the global leaders ultimately governs the direction of the search. The swarm should efficiently navigate the design space such that the search effort is locally focused within the preferred region and provides a uniform spread of solutions. Since all members of the archive are mutually non-dominated, a ranking procedure is necessary to distinguish the most appropriate candidates for leadership from the remaining members. At each time-step t, the most preferred solution $\mathbf{z}'(t)$ is recorded. The subset of members $\mathbf{X}_g(t) \in \mathbf{Q}(t)$ selected for global leadership satisfy the condition of Eq. (5.3), such that

$$\mathbf{X}_g(t) \in \mathbf{Q}(t) : \; d_z(\mathbf{Q}(t)) \leq d_z(\mathbf{z}'(t)) + \delta. \tag{5.6}$$



(a) Leader selection via Eq. (5.6)



(b) Leader selection via Eq. (5.7)

Figure 5.6: Defining swarm leaders based on global leader distance-based schemes

Since not every member will initially satisfy this condition, the number of candidate leaders may fluctuate over time. This condition provides the necessary selection pressure for particles to locally focus the search effort within the preferred region, avoiding the unnecessary computational effort of exploring undesired regions of the design space. Each swarm particle is randomly assigned a leader to promote diversity in the search. Figure 5.6(a) illustrates the selection of leaders based on the preferred region concept.

In the case where all non-dominated solutions satisfy the condition of Eq. (5.6), additional guidance through the crowding distance metric is provided. Members are first sorted based on their crowding distance values $d_c()$. The top 10% ranked members according to $d_c()$ (i.e. members with the highest crowding distance values) are subsequently selected for global leadership, to promote a uniform spread. Figure 5.6(b) shows an instance where

Algorithm 3 Truncating the archive

```
1: if K > K_max then
2:    if max d_z(Q) > d_z(z' + δ) then
3:       q_max(d_z) = x_i
4:    else
5:       q_min(d_c) = x_i
6:    end if
7: end if
```

it is desirable to focus on solutions in sparsely populated areas. The definition of global leaders is therefore governed as,

$$\mathbf{X}_g(t) \in \mathbf{Q}(t) : \ \text{sort}(d_c(\mathbf{Q}), \max) \ \ \textbf{if} \ \ \max d_z(\mathbf{Q}) \leq d_z(\mathbf{z}' + \delta). \tag{5.7}$$

As the particles are guided to converge to the preferred region, the number of identified non-dominated solutions will steadily increase. To avoid this number unnecessarily escalating and to maintain high competitiveness within the archive, there is a restriction (denoted by $K_{\max}$) on the number of solutions permitted for entry. If the number of members $K > K_{\max}$, the newest solution is permitted entry and an existing member is removed. If all archive members exist within the preferred region, the most crowded solutions are removed. This ensures that solutions in densely populated regions are removed in favour of solutions which exploit sparsely populated regions, to promote a uniform spread. For each new addition $\mathbf{x}_i$, an existing member $\mathbf{q}$ is removed as shown in Algorithm 3.

5.2.2 Personal Best Selection Scheme

Fieldsend and Singh [2002] proposed a population of personal best solutions for each particle i, with the best representative front found by the solution i. While the benefit could be greater search diversity, the expense of maintaining N personal archives is quite high and the same degree of diversity could potentially be achieved by an appropriate global selection mechanism or through a turbulence operator [Fieldsend, 2004].

Most of the other reported MOPSO variants [Fieldsend, 2004] maintain one personal best solution $\mathbf{p}_i$ for each particle i, which is updated during the search according to the dominance criteria. The ambiguity lies in the treatment of the case when the personal best solution $\mathbf{p}_i(t)$ is mutually non-dominated with the solution $\mathbf{x}_i(t+1)$. The introduction of the reference point distance metric elegantly deals with this ambiguity. If the particle position $\mathbf{x}_i(t+1)$ is mutually non-dominated and preferred to the existing personal best $\mathbf{p}_i(t)$, then the personal best is replaced. Otherwise the personal best is remained unchanged. The personal best of particle i is updated as per Algorithm 4.

Algorithm 4 Updating the personal best position

```
for each particle x_i do
  if x_i ≺ p_i then
    p_i = x_i
  else if x_i ⪯ p_i then
    if d_z(x_i) < d_z(p_i) then
      p_i = x_i
    end if
  end if
end for
```

5.2.3 Constraint Handling

Constraints in PSO may be handled in different ways. Particles may simply be rejected if they violate a constraint, or a penalty factor could be applied to persuade other particles from exploring that region of the design space [Engelbrecht, 2005]. Deb *et al.* [2002] introduced a constraint-handling mechanism based on Pareto ranking, which applies specifically to MOO. The dominance criterion is manipulated to include the feasibility of a solution with respect to its number of constraint violations. For the i-th particle, $\mathbf{c}(\mathbf{x}_i) = \{c_1, \ldots, c_p\}$ where p is the number of constraints and $\mathbf{c}(\mathbf{x}_i) \geq 0$ is the violation of the constraint. A particle $\mathbf{a}$ therefore constraint-dominates $\mathbf{b}$ if any of the following criteria are met:

- Solution $\mathbf{a}$ is feasible and solution $\mathbf{b}$ is not.
- Both solutions $\mathbf{a}$ and $\mathbf{b}$ are infeasible but $\mathbf{c}(\mathbf{a}) < \mathbf{c}(\mathbf{b})$.
- Both solutions $\mathbf{a}$ and $\mathbf{b}$ are feasible but $\mathbf{a} \prec \mathbf{b}$.

If both particles are infeasible, the particle with the overall least constraint violation is considered the better solution. If both particles are deemed to satisfy the constraint values, then evidently the better performing solution is admitted for entry. This methodology alleviates the ambiguity in having to define a penalty factor.

5.2.4 Mutation Operator

The concept of mutation in EA as a genetic operator was introduced in Section 3.2.2. For single point mutation, the candidate genome is mutated by flipping a single binary bit. Mutation is an important operator since it reduces the probability of premature convergence to inferior solutions, and provides a mechanism to maintain diversity within the population. For PSO, the notion of mutation and crossover is inherent within the

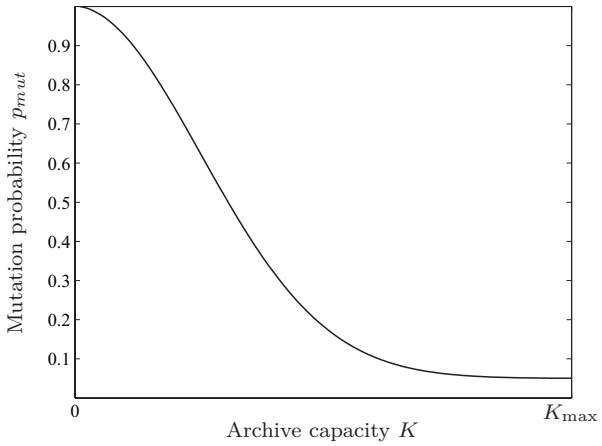


Figure 5.7: Mutation probability p_{mut} as a function of archive capacity

particle update equations. However for reasons of avoiding premature convergence, there are many reported PSO algorithms in the literature which incorporate explicit mutation schemes [Engelbrecht, 2005; Reyez-Sierra and Coello Coello, 2006]. Despite the additional guidance provided by the reference point, and the diversity inherent within the proposed topology, the search proficiency of the swarm may deteriorate when confronted with a highly multi-modal problem. It follows that as the velocity of the particles converges to zero (i.e. $\mathbf{v} \rightarrow 0$), the swarm is unable to generate new leaders. This could lead to premature convergence as a result of the swarm being trapped within a local front. Kennedy and Eberhart [1995] observed that the probability of such an occurrence may be reduced by introducing some element of flight turbulence or craziness. Turbulence, or mutation, issues a random variation in the particle flight trajectory or position.

Reyez-Sierra and Coello Coello [2006] present a survey of MOPSO algorithms, also characterizing their turbulence or mutation schemes. A mutation operator (when applied appropriately) is very effective at generating new leaders. If mutation is incorrectly implemented, it becomes destructive and deteriorates the natural explorative capabilities of the swarm. Generally the mutation operator in reported algorithms is applied directly to particles in the swarm, via a mutation probability criterion [see Reyez-Sierra and Coello Coello, 2005; Wickramasinghe and Li, 2008]. For the proposed algorithm, a Gaussian mutation operator is applied to members within the archive. Mutation is triggered if consistent improvement in the archive is not recorded[1]. To ensure that the mutation operator is non-destructive, only mutated members which provide some measure of improvement over existing members are successful. The percentage of archive members p_{mut} selected for mutation steadily reduces as the archive reaches maximum capacity, as shown in Fig-

[1]Improvement is measured by monitoring the mean value of d_z. If successive mean values are equal, mutation is triggered.

Algorithm 5 Gaussian mutation operator scheme

```
for each archive member q_{i...K} do
  if rand ≤ p_mut then
    for each dimension j do
      if rand ≤ p_dim then
        MUTATE dimension q'_{i,j}
      end if
    end for
    if d_z(q') < max d_z(Q) then
      UPDATE archive q_{K+1} = q'_i
    end if
  end if
end for
```

ure 5.7. A mutation probability of $p_{dim} = 0.1$ is applied to each dimension. For the selected member $\mathbf{q}_i$ and dimension j, Gaussian mutation is applied as follows:

$$q'_{i,j} = q_{i,j} + N(\mu, \sigma^2) \tag{5.8}$$

where $N(\mu, \sigma^2)$ represents a random number of mean $\mu = 0$ generated from the normal distribution curve. The standard deviation, σ^2 (based on trial and error) is given a value of 0.1 which plateaus to 0.05 as the algorithm advances to promote exploitation. Theoretically the mutation operator should only come into effect if the inherent searching capabilities of the swarm fail – thereby resulting in premature convergence to a local front. Algorithm 5 illustrates the operation of the mutation scheme.

The global leader selection scheme introduced in Section 5.2.1 incorporates the reference point distance metric and dominance to promote solution diversity. The mutation operator however is used solely to enhance diversity. No reference is made to dominance, and improvement is solely measured by preference comparison. This proves to be an effective technique for establishing global leaders in multi-modal optimization problems.

5.2.5 UPMOPSO Pseudo Code

The User-Preference Multi-Objective Particle Swarm Optimization (UPMOPSO) algorithm is summarized in Algorithm 6. The algorithm utilizes the swarm topology, velocity initialization and boundary violation schemes as described in Section 3.3. The constriction *type 1* formulation of Clerc and Kennedy [2002] is implemented to update the velocity and position of each particle, as described in Section 3.3.3. The stopping criterion is either based on the maximum number of time-steps $t_{\max}$ or the maximum number of function evaluations $f_{\max}$ as specified by the user.

Algorithm 6 The UPMOPSO algorithm

```
OBTAIN user-defined preferences, z and δ
INITIALIZE swarm of size [N, n]; t = 0
EVALUATE fitness and distance metric
ASSIGN personal best
CONSTRUCT archive Q(t + 1)
t = 1
repeat
    SELECT global leaders
    UPDATE particle velocity
    CONSTRICT velocity
    UPDATE particle position
    APPLY boundary conditions
    EVALUATE fitness and distance metric
    UPDATE personal best
    UPDATE archive Q(t + 1)
    if MUTATION triggered then
        MUTATE archive members
    end if
    t = t + 1
until t = t_max OR f_max
```

5.3 Test Function Validation

To illustrate the functionality of the UPMOPSO algorithm, a series of test functions are presented. Multi-objective test function suites have been utilized in order to demonstrate the performance of the algorithm for problems of varying size and complexity. Details of each test function are provided in Appendix A. The behaviour of the algorithm during exploration and exploitation is highlighted, and guidelines to controlling the spread of solutions and objective bias are demonstrated through selection and manipulation of the user-preference parameters. A modified hyper-volume metric is introduced to monitor convergence and solution spread, which does not depend on prior knowledge of the Pareto front. To facilitate the performance comparison, results of the NSGA-II [Deb *et al.*, 2002] and the reference point based R-NSGA-II [Deb and Sundar, 2006] are also included.

5.3.1 Convergence Characteristics

A simple execution of the UPMOPSO algorithm is provided in Figure 5.8 on the two-objective ZTD1 test function [Zitzler *et al.*, 2000]. A swarm of $N = 100$ particles is initialized and flown for 50 time-steps. The reference point is selected as $\bar{\mathbf{z}} = [0.5, 0.5]$ with a solution spread of $\delta = 5 \times 10^{-2}$. The archive solutions are recorded and plotted at

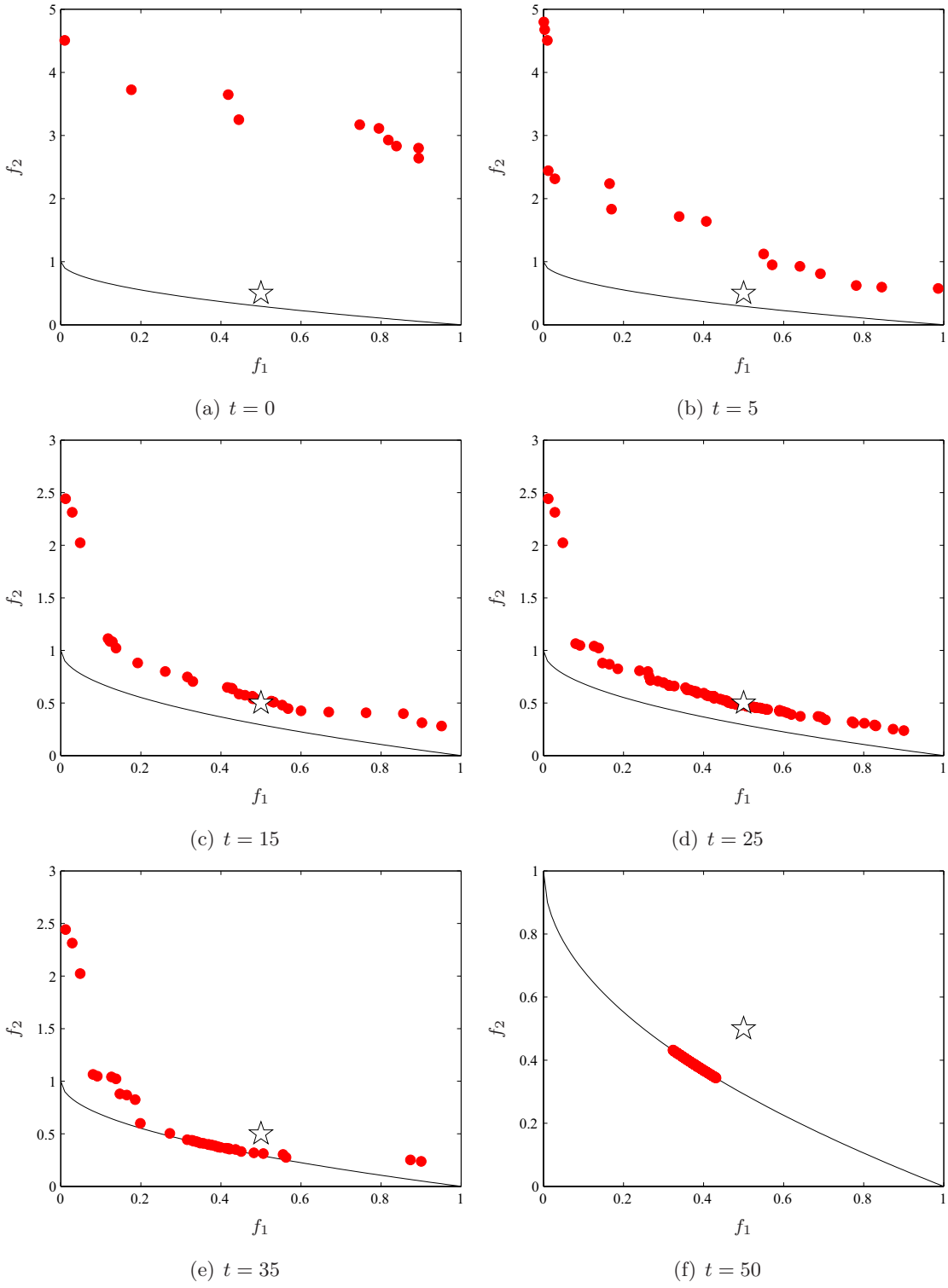


Figure 5.8: Illustration of the UPMOPSO algorithm on the ZTD1 two-objective test function, showing phases of initialization, exploration and exploitation

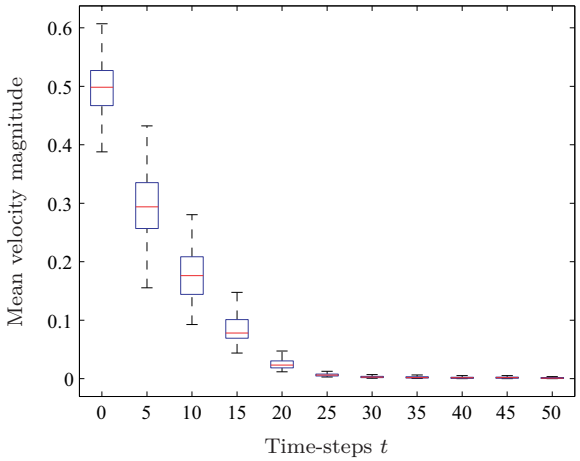


Figure 5.9: Statistical box-plots illustrating particle stagnation due to convergence

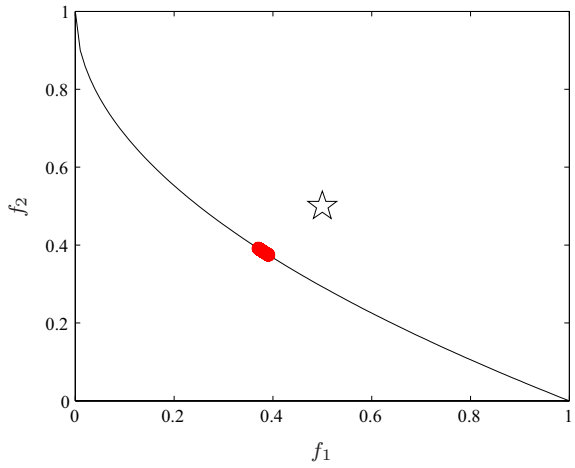
different iterations, to highlight the phases of exploration and exploitation. The swarm is initialized in Figure 5.8(a), and the exploration phase is evident from Figures 5.8(b) and 5.8(c). A distinct attraction towards the preferred region of the Pareto front is shown thereafter. The swarm is able to exploit this region to provide a uniform set of non-dominated solutions as shown in Figure 5.8(f).

Figure 5.8 illustrates the convergence of the swarm to the Pareto front by monitoring the activity of the archive members. While this methodology effectively communicates the proficiency of the swarm in identifying non-dominated solutions, the truest reflection of convergence is by monitoring the particle dynamics. Figure 5.9 shows box-plots of the mean velocity magnitude for each particle. A clear convergence towards *stagnant motion* is observed, indicating the arrival of the swarm to the preferred area of the design space.

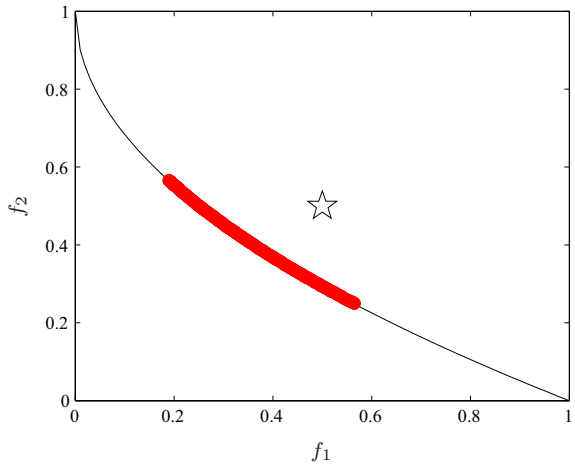
5.3.2 Controlling the Preferred Region

Isolating and identifying the preferred region on the Pareto front is essentially dependent on the parameters $\bar{\mathbf{z}}$ and δ. The δ parameter is fairly straightforward, as it essentially dictates the extent of the solution spread. Figure 5.10 illustrates the convergence of the UPMOPSO algorithm on the ZTD1 test function, with varying δ.

While the concept of the reference point $\bar{\mathbf{z}}$ may initially seem misleading, it in fact provides a very simple and intuitive method of articulating the preferred interests of the designer. Consider the three-objective DTLZ2 test function [Deb *et al.*, 2005], where the boundaries of the Pareto front are in the unit hypercube $\mathbf{f} \in [0, 1]^3$. If an absolutely equal weighting between all objectives is desired, then a reference point consisting of equal values should be considered, as illustrated in Figure 5.11(a). This is however not dependent on the feasibility of the reference point, as illustrated in Figure 5.11(b).



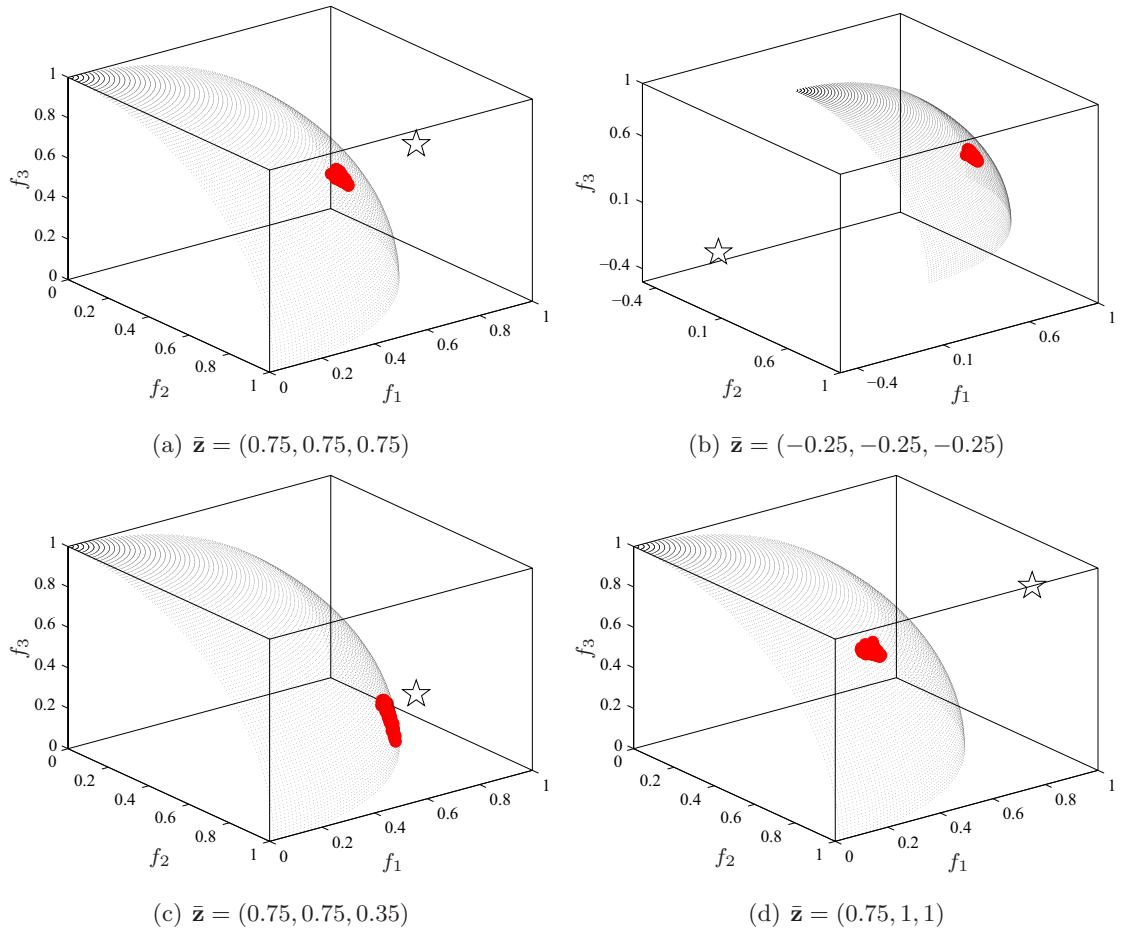
(a) $\delta = 1 \times 10^{-2}$



(b) $\delta = 2 \times 10^{-1}$

Figure 5.10: Controlling the extent of the solution spread by varying δ

There are a few possibilities to provide additional bias to any particular objective(s). For example, the simplest approach to applying additional weighting to the objective f_3 would be to reduce the reference value for f_3, as illustrated in Figure 5.11(c). Alternatively it is possible to obtain a similar level of bias to a specific objective by relaxing the reference values of the other objective functions. This is illustrated in Figure 5.11, where the values of f_2 and f_3 are relaxed, naturally placing more emphasis on the objective f_1. It is demonstrated that by adding additional bias to any particular objective, the preferred region shifts to favour that objective without affecting the compromise between any other objectives. In the context of aerodynamic design, if a hypothetical existing design configuration were representative of an original reference point, then a new reference point which emphasizes a specific design condition represents a *target* compromise where the designer is interested in obtaining further improvement in that specific design objective.



(a) $\bar{\mathbf{z}} = (0.75, 0.75, 0.75)$ (b) $\bar{\mathbf{z}} = (-0.25, -0.25, -0.25)$

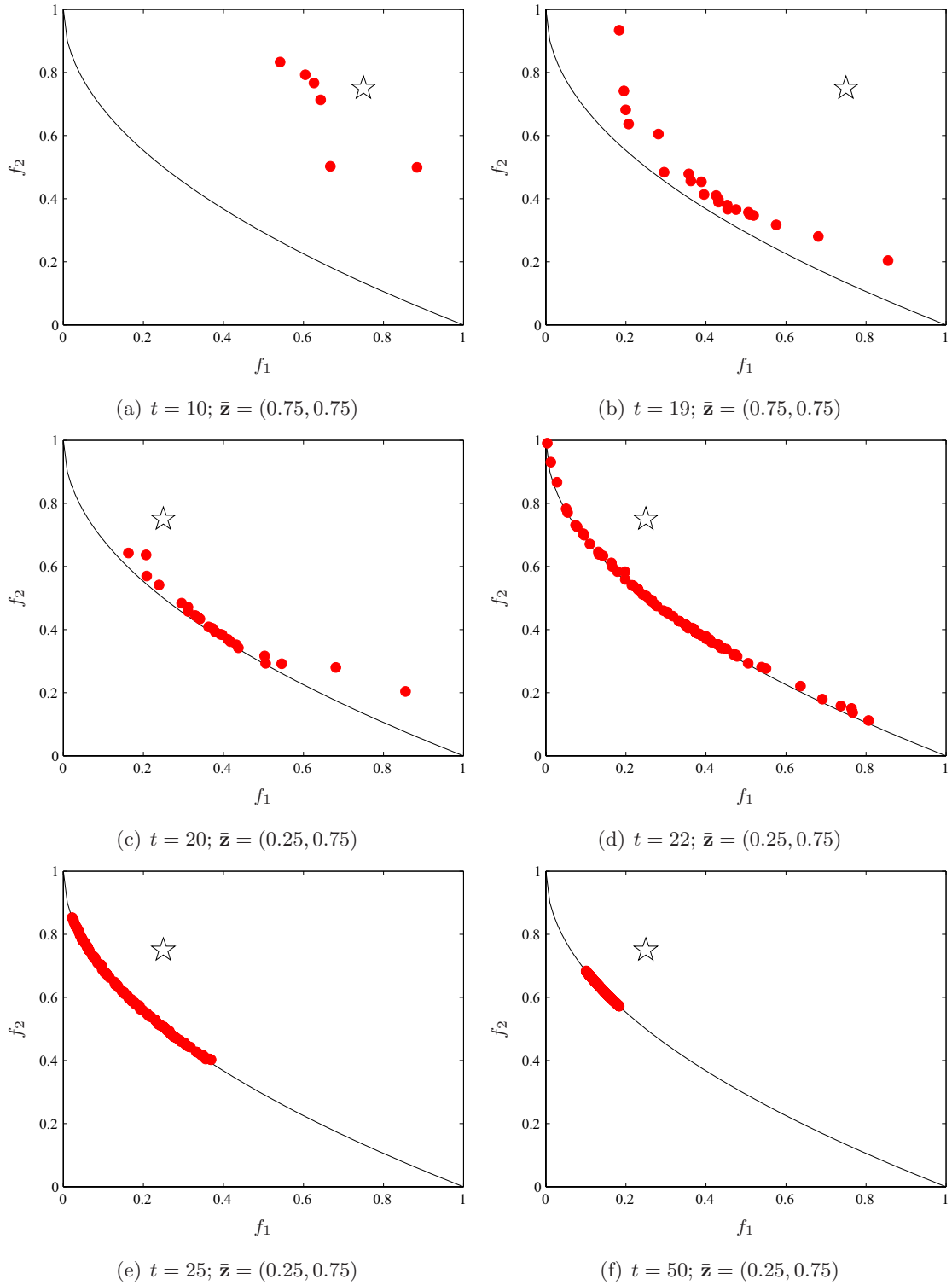
(c) $\bar{\mathbf{z}} = (0.75, 0.75, 0.35)$ (d) $\bar{\mathbf{z}} = (0.75, 1, 1)$

Figure 5.11: Shifting the preferred region through adjustment of the reference point

The example given in Figure 5.11 is indicative of the scenario when the designer places emphasis on a particular objective before the optimization process. It is however possible that during the course of optimization it may be necessary to shift the preferred region without the added computational expense of reinitializing the swarm. Essentially the search direction of the swarm (and consequently the identified preferred region) can simply be disrupted or modified at any time-step during flight by manipulating the reference point. To illustrate this concept, Figure 5.12 shows the interval convergence of a swarm of $N = 100$ particles flown for 50 time-steps, on the ZTD1 test function. At time $t = 20$, the reference value $\bar{z}_1$ is reduced, thereby placing more emphasis on the first objective. Despite the abrupt change in search direction, the swarm quickly adapts to the change in preferences and subsequently converges to the updated preferred region.

5.3.3 Hyper-Volume Performance Metric

There are several popular performance metrics to monitor solution spread and accuracy for MOO [see Zitzler *et al.*, 2000] yet these metrics are only suitable for problems where



(a) $t = 10$; $\bar{\mathbf{z}} = (0.75, 0.75)$

(b) $t = 19$; $\bar{\mathbf{z}} = (0.75, 0.75)$

(c) $t = 20$; $\bar{\mathbf{z}} = (0.25, 0.75)$

(d) $t = 22$; $\bar{\mathbf{z}} = (0.25, 0.75)$

(e) $t = 25$; $\bar{\mathbf{z}} = (0.25, 0.75)$

(f) $t = 50$; $\bar{\mathbf{z}} = (0.25, 0.75)$

Figure 5.12: Illustration of the UPMOPSO algorithm on the ZTD1 two-objective test function, showing the mid-flight shift in the preferred region due to an adjustment of the reference point

the Pareto front is known. For design problems, the Pareto front is generally unknown and a suitable performance metric is therefore required which does not rely on this information to compare various algorithms. The Hyper-Volume (HV) metric is a popular performance metric, which provides a single measurement to assess both the convergence and spread of solutions and does not rely on knowledge of the Pareto front [Knowles and Corne, 2002]. The HV performance metric is applied to the archive members $\mathbf{Q} = \{\mathbf{q}_1, \cdots, \mathbf{q}_K\}^T$. The HV metric calculates the total volume bounded by the archive solutions on the Pareto front and a selected point in the objective-space. The selected point is termed the *nadir point*. At the nadir point, all objectives are at their worst values simultaneously [Emmerich *et al.*, 2005]. The nadir point $\mathbf{q}_{nad}$ is calculated by first collating all identified solutions (from all algorithms), such that

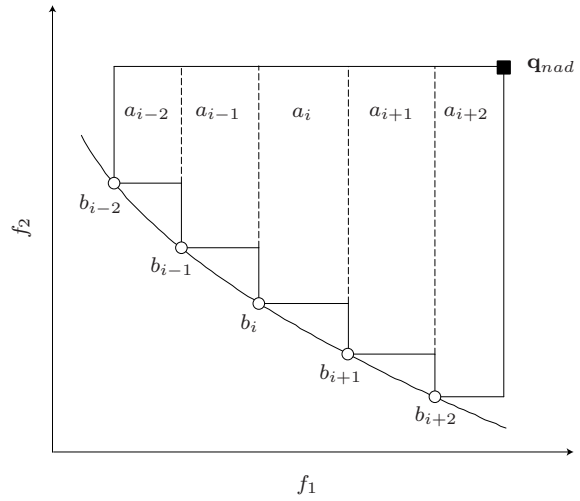
$$\mathbf{q}_{nad} = [f_1^{nad}(\mathbf{q}), \ldots, f_M^{nad}(\mathbf{q})] \quad \text{where} \quad f_i^{nad}(\mathbf{q}) = \max_{j=1\ldots K} \{f_i(\mathbf{q}_j)\}. \tag{5.9}$$

The formal definition of the HV metric according to Emmerich *et al.* [2005] is the Lebesgue measure, Λ, of the union of all hypercubes a_i defined by a solution $\mathbf{b_i}$ and the nadir point $\mathbf{q}_{nad}$. The HV value is the sum of all these volumes as denoted in Figure 5.13(a).

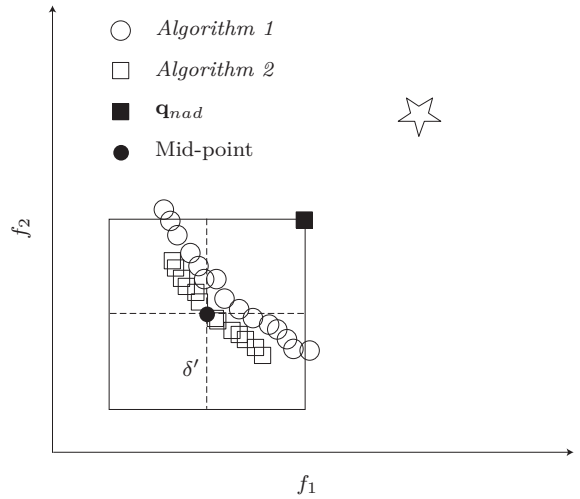
$$HV(\mathbf{Q}) = \Lambda(\{\bigcup_i a_i | \mathbf{b}_i \in \mathbf{Q}\}) = \Lambda(\bigcup_{\mathbf{b}_i \in \mathbf{Q}} \{\mathbf{q} | \mathbf{b}_i \prec \mathbf{q} \prec \mathbf{q}_{nad}\}) \tag{5.10}$$

For experimental validation with stochastic algorithms, multiple runs are conducted to eliminate the influence of random occurrences. The nadir point is calculated from the final combined population, and is subsequently used to obtain the HV value for each independent simulation. When comparing different algorithms, that which provides the greatest HV value is considered to be superior, since it gives a measure of both the spread and the closeness of the solution to the Pareto front.

The standard HV metric is not applicable for comparing preference-based optimization algorithms since the search converges to a partial subset of the Pareto front. A modification to the standard HV calculation was proposed by Wickramasinghe *et al.* [2010]. All solution points outside the preferred regions are excluded such that the nadir point used to calculate the HV metric is representative of the preferred region. After each independent simulation, all non-dominated solutions are combined to produce one non-dominated set. Wickramasinghe *et al.* then suggest that the Euclidean distance is calculated of each solution to a fictitious ideal point (which for a minimization problem would be placed at the origin). A volume is defined around the solution point with the lowest Euclidean distance to the ideal point. The value δ' is prescribed such that a volume is defined having $2\delta'$ for each objective. The value of δ' may be adjusted such that a sufficient number of solution points lie within the defined volume. Solutions which reside outside the preferred volume



(a) Definition of hypercubes



(b) Volume boundary via definition of the mid-point

Figure 5.13: The modified HV performance metric to monitor convergence and solution spread

are simply excluded from the calculation. The remaining solutions are used to obtain the nadir point, which is located within the defined volume. The calculation of the HV values for each independent run is then performed using this common nadir point. This original method has one inconsistency, since the ideal point is dependent on whether the volume is at the centre or either extreme of the front. The same procedure is utilized in this research, with the exception that the original ideal point is no longer used to define the boundary, but rather the mid-point solution of the preferred region.

5.3.4 Test Functions

Presented in this section are the results of UPMOPSO for various test problems obtained from MOO test function suites. It is sought to provide an unbiased overview of the

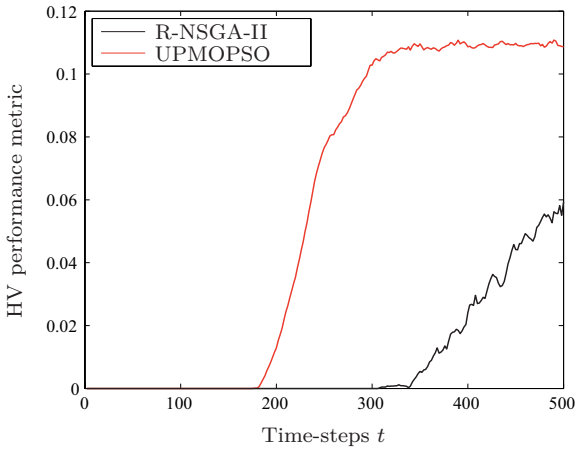


Figure 5.14: Time-step convergence for the multi-modal ZTD4 test problem

capabilities of the algorithm for problems of varying size and complexity. Results of both the NSGA-II and the reference point based R-NSGA-II algorithms are included for performance comparison. The HV metric is used to quantify the performance of the algorithms, averaged over ten independent runs to eliminate random occurrences.

ZTD Test Suite

Zitzler *et al.* [2000] proposed a test suite of six bi-objective minimization problems. The detailed descriptions of these test functions are provided in Appendix A. In the ZTD test function suite, $x_i \in [0, 1]$ for all variables, apart from the ZDT4 function where $x_1 \in [0, 1]$ and $x_i \in [-5, 5]$ for all other variables. The functions ZTD1–3 and 6 are biased two objective problems, hence the global front is fairly simple to identify. The ZTD4 problem is however highly multi-modal, and consists of 21^9 local inferior fronts. The binary encoded ZTD5 problem has been omitted.

Properties of the ZTD test problems are provided in Table 5.1. Each test problem is assigned a single reference point and solution spread value to compute the HV metric. A swarm of $N = 100$ particles is initialized for each problem instance and flown for the prescribed number of time-steps. This corresponds to a population of 100 individuals evolving for the maximum number of generations for the GA equivalents. The mutation operator of the UPMOPSO algorithm is included for the ZTD4 test function, as the algorithm would otherwise not converge to the Pareto front. Results of the computational experiments are provided in Table 5.1.

The ZTD test function suite is fairly biased towards the Pareto front (apart from the ZTD4 problem) and the UPMOPSO algorithm converges to the Pareto front within 50 to 100 time-steps. For ease of comparison and consistency between results, the maximum

Problem	n	**Characteristics**	$t_{\max}$	**NSGA-II**	**R-NSGA-II**	**UPMOPSO**
ZTD1	30	Convex	200	0.0755 ± 0.0129	0.1055 ± 0.0046	**0.1101 ± 0.0020**
ZTD2	30	Concave	200	0.0892 ± 0.0107	0.1127 ± 0.0052	**0.1206 ± 0.0026**
ZTD3	30	Discontinuous	200	0.3638 ± 0.1142	0.4436 ± 0.0179	**0.4601 ± 0.0073**
ZTD4	10	Convex	500	0.0490 ± 0.0240	0.0628 ± 0.0215	**0.1112 ± 0.0098**
ZTD6	10	Concave	200	0.0721 ± 0.0080	0.0910 ± 0.0070	**0.0975 ± 0.0023**

Table 5.1: Properties and average HV results of the ZTD test function suite. Results of the highest preforming algorithm are shown in bold.

Problem	n	**Characteristics**	$t_{\max}$	**NSGA-II**	**R-NSGA-II**	**UPMOPSO**
DTLZ1	7	Linear	500	0.1436 ± 0.0242	0.1967 ± 0.0260	**0.2326 ± 0.0121**
DTLZ2	12	Concave	200	0.2915 ± 0.0852	**0.4082 ± 0.0134**	0.3863 ± 0.0210
DTLZ3	12	Concave	500	0.0000 ± 0.0000	0.3765 ± 0.0218	**0.3922 ± 0.0207**
DTLZ7	22	Discontinuous	100	0.0372 ± 0.0785	0.0945 ± 0.0173	**0.1112 ± 0.0120**

Table 5.2: Properties and average HV results of the three-objective DTLZ test function suite. Results of the highest preforming algorithm are shown in bold.

number of time-steps is fixed to 200. Table 5.1 highlights the superiority of the UPMOPSO algorithm over the NSGA variants, especially for the highly multi-modal ZTD4 function. Figure 5.14 demonstrates the time-step convergence of the UPMOPSO algorithm for the ZTD4 test function. It is demonstrated that the UPMOPSO algorithm is superior to the preference-based R-NSGA-II algorithm in both phases of exploration and exploitation.

DTLZ Test Suite

Deb *et al.* [2005] proposed the DTLZ suite as an extension to the ZTD test function suite. The detailed descriptions of these test functions are provided in Appendix A. The unique feature of the DTLZ test suite (at the time) is that the problems may be scaled to any number of objectives and dimensions. In the DTLZ test function suite, $x_i \in [0, 1]$ for all

Problem	n	**Characteristics**	$t_{\max}$	**NSGA-II**	**R-NSGA-II**	**UPMOPSO**
DTLZ1	9	Linear	500	0.0000 ± 0.0000	0.2454 ± 0.0581	**0.3157 ± 0.0132**
DTLZ2	14	Concave	200	0.0000 ± 0.0000	0.3445 ± 0.0416	**0.3757 ± 0.0181**
DTLZ3	14	Concave	500	0.0000 ± 0.0000	0.3610 ± 0.0198	**0.3755 ± 0.0117**

Table 5.3: Properties and average HV results of the five-objective DTLZ test function suite. Results of the highest preforming algorithm are shown in bold.

variables. DTLZ1 is a multi-modal problem, consisting of $11^5 - 1$ local optimal fronts. DTLZ3 is highly multi-modal, with $3^{10} - 1$ local optimal fronts. The mutation operator of the UPMOPSO algorithm is included for these two test functions, as the algorithm is otherwise not guaranteed to converge to the Pareto front. The DTLZ2 and DTLZ7 test problems also included in this thesis are biased problems.

Properties and results of the DTLZ test problems is shown in Table 5.2 for three-objective variants, and in Table 5.3 for five-objective variants. Each test problem is assigned a single reference point and solution spread value to compute the HV metric. A swarm of $N = 100$ particles is initialized for each problem instance and flown for the prescribed number of time-steps. This corresponds to a population of 100 individuals evolving for the maximum number of generations for the GA equivalents.

UPMOPSO outperforms the NSGA-II variants for all test problems, apart from the three-objective DTLZ2 problem where the R-NSGA-II algorithm is superior. As the scale of the problem increases the benefits of adopting user-preferences are more apparent as in some cases NSGA-II is not able to converge to the Pareto front. UPMOPSO consistently produces higher average HV values and lower standard deviation values, which highlights its proficiency in providing an accurate and uniform approximation of the preferred region. Shown in Figures 5.15 and 5.16 are solution fronts of certain problems.

Constrained Problems

A series of problems are included to asses the performance of the UPMOPSO algorithm in handling (conditional) constrained problems. Included in this thesis are the constrained problems of Kita *et al.* [1996] (KITA) and Osycza and Kundu [2004] (OSY), as well as the two-bar truss and welded beam design problems of Deb [2000]. Details of these problems are provided in the Appendix. The problems are not difficult or multi-modal, however the constraints are of varying size and complexity.

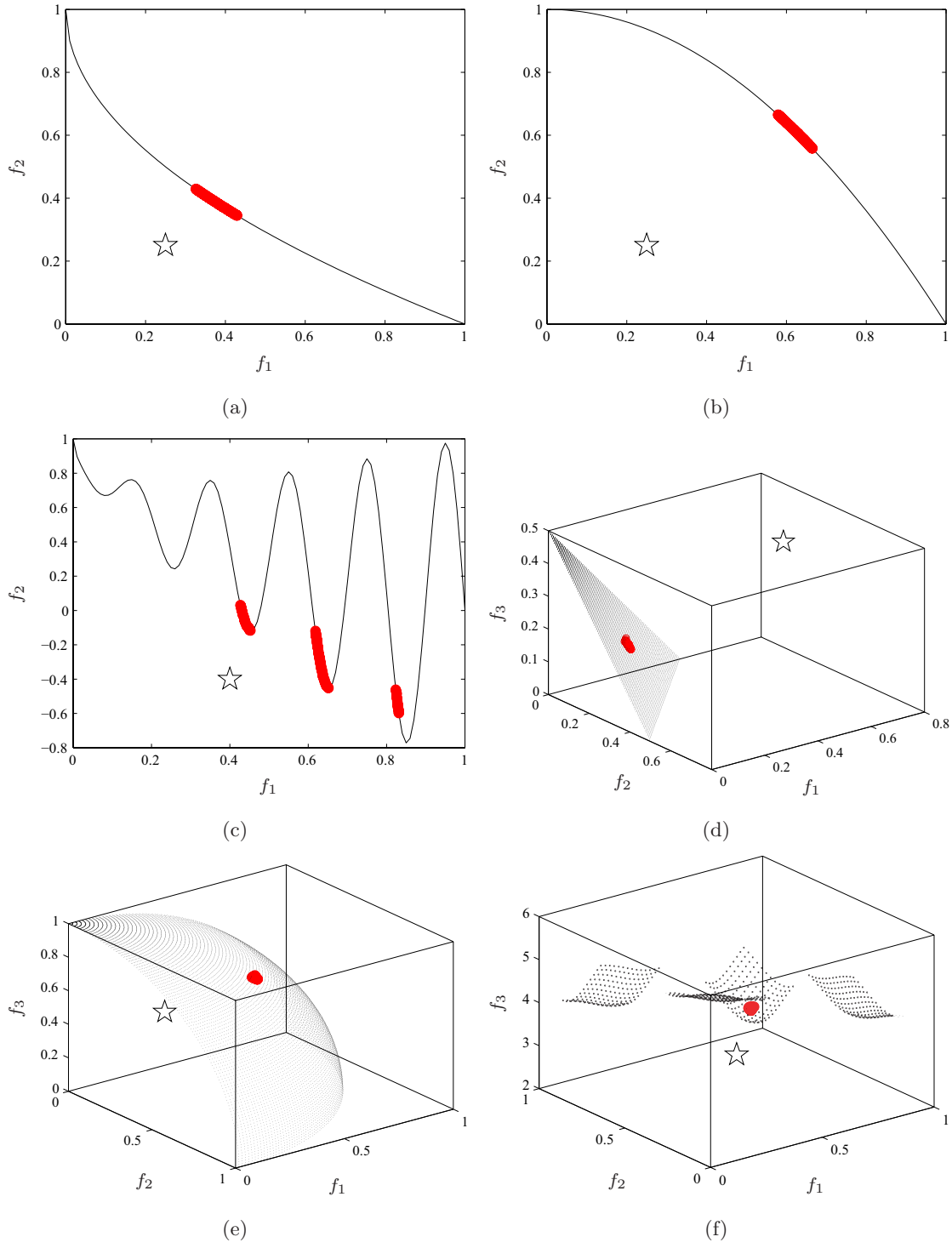
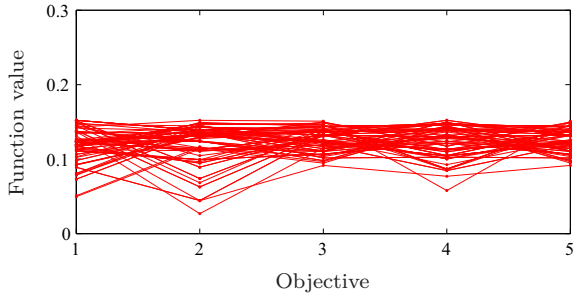
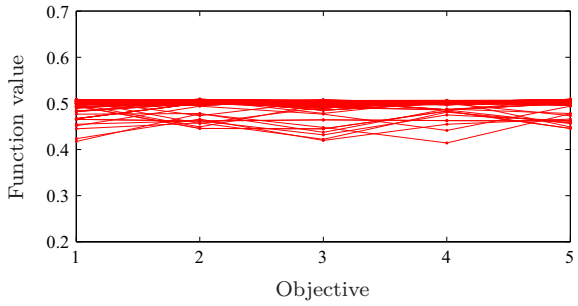


Figure 5.15: Solution front as obtained by the UPMOPSO algorithm with $\bar{z}_i = 0.5$ for the (a) ZTD4; (b) ZTD2; (c) ZTD3; (d) three-objective DTLZ1; (e) three-objective DTLZ2; and (f) three-objective DTLZ7 problems



(a)



(b)

Figure 5.16: Solution front as obtained by the UPMOPSO algorithm for the (a) five-objective DTLZ1; and (b) five-objective DTLZ3 problems

Problem	n	**Constraints**	**NSGA-II**	**R-NSGA-II**	**UPMOPSO**
KITA	2	3	0.2792 ± 0.0943	0.3384 ± 0.0663	**0.3866 ± 0.0241**
OSY	6	6	0.4591 ± 0.0976	0.5518 ± 0.0577	**0.5712 ± 0.0491**
Two-bar truss	3	1	0.0932 ± 0.0452	0.1453 ± 0.0230	**0.1529 ± 0.0137**
Welded beam	4	4	0.5586 ± 0.1037	0.6076 ± 0.0663	**0.6245 ± 0.0513**

Table 5.4: Properties and average HV results of the constrained test problems. Results of the highest preforming algorithm are shown in bold.

Properties and results of the constrained test problems are shown in Table 5.4. Each problem is characterized by two objectives, and the number of variables and constraints are indicated. Each test problem is assigned a single reference point and solution spread value to compute the HV metric. A swarm of $N = 100$ particles is initialized for each problem instance and flown for a maximum number of time-steps of 100. The solution fronts are shown in Figure 5.17. Results clearly demonstrate that the benefits of focusing exclusively within a preferred region of the Pareto front also translate to constrained problems.

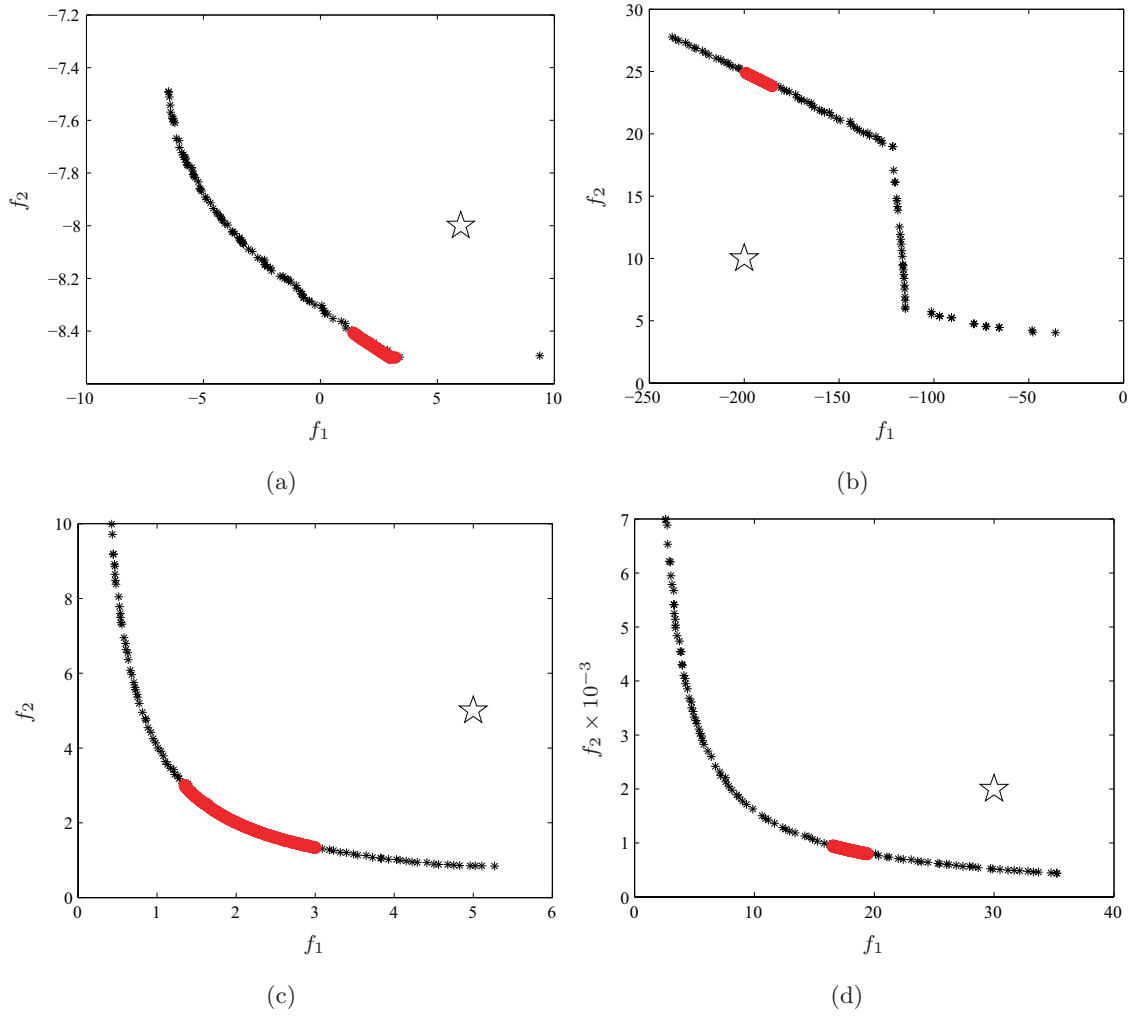


Figure 5.17: Solution front as obtained by the UPMOPSO (•) and NSGA-II (*) algorithms for the (a) KITA; (b) OSY; (c) Two-bar truss; and (d) Welded beam constrained problems

5.3.5 Low-fidelity Design: Wave Drag Optimization

To further demonstrate the operation of the UPMOPSO algorithm, a low-fidelity aerodynamic design case-study is presented. This study demonstrates the relative ease in assigning a reference point to design problems to efficiently identify preferred designs. This example deals with a wave drag airfoil shape optimization problem that aims to minimize $f_1 = C_d/C_l^2$ and $f_2 = C_m^2$, for an angle of incidence of $\alpha = 2^\circ$ and a Mach number of $M = 0.73$. This formulation has appeared in earlier studies, notably in Vicini and Quagliarella [1997] and Ray and Tsai [2004].

The PARSEC method (as described in Section 2.2.3) is used to parameterize the airfoil geometry, and the potential flow solver (as described in Section 2.4.3) is selected to predict the aerodynamic coefficients. The potential flow solver is not computationally intensive

Variable	Lower Bound	Upper Bound
r_{LE}	0.0055	0.0085
α_{TE}	$-12°$	$-8°$
β_{TE}	9.5°	14.5°
x_{UP}	0.3	0.5
z_{UP}	0.05	0.075
$z_{xx_{UP}}$	-0.6	-0.4
x_{LO}	0.28	0.42
z_{LO}	-0.075	-0.5
$z_{xx_{LO}}$	0.55	0.85
z_{TE}	0	0
Δz_{TE}	0	0

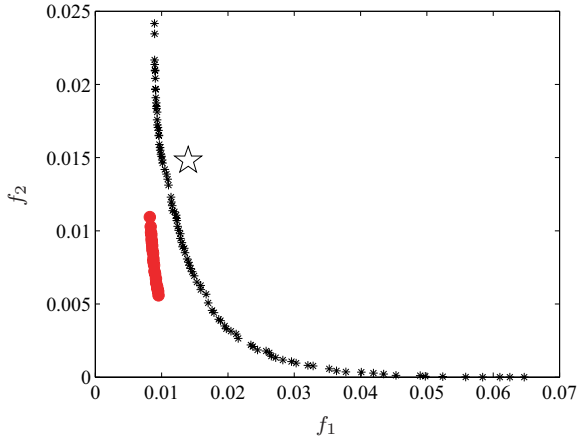
Table 5.5: PARSEC parameter ranges for the low-fidelity wave drag optimization study

and requires only a few seconds per function evaluation, yet still captures important shock-flow phenomena. The boundaries of the design space (see Table 5.5) are as suggested by Ray and Tsai [2004]. The reference point is selected as the benchmark RAE2822 airfoil, since it provides maximum performance within the desired operating range. The reference values as obtained by the potential flow solver are $\bar{\mathbf{z}} = [0.0141, 0.0148]$.

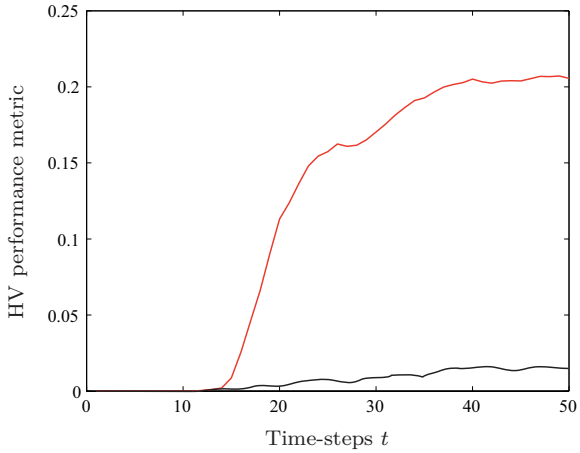
A swarm population of $N = 100$ particles is initialized and flown for $t_{\max} = 50$ time-steps. Due to the apparent simplicity of the optimization problem, the mutation operator is omitted. While it is expected that the optimizer will converge before the maximum allocated number of time-steps, the process is continued to highlight the phases of exploration and exploitation. For comparative purposes, the NSGA-II algorithm is also applied to obtain the full Pareto front. Figure 5.18 shows the final solution sets obtained by the UPMOPSO and NSGA-II algorithms, accompanied with the HV convergence plots.

It is observed from Figure 5.18(a) that the reference point is dominated with respect to the final solution spread. The UPMOPSO algorithm is therefore able to identify designs which provide improvement in all objectives, while reflecting the reference point compromise. Figure 5.18(a) also demonstrates that the UPMOPSO algorithm provides a more optimal solution set than NSGA-II, which highlights its superiority in exploitation, no doubt an advantage of focusing exclusively on a subset of the Pareto front. This is further confirmed in Figure 5.18(b), which conclusively demonstrates the greater convergence characteristics of the UPMOPSO algorithm in both exploration and exploitation.

Table 5.6 shows the designs of interest of the optimization case-study, as identified by the UPMOPSO algorithm. The preferred design $\mathbf{z}'$ clearly provides a design which reflects



(a) Solution fronts of NSGA-II (*) and UPMOPSO (•)



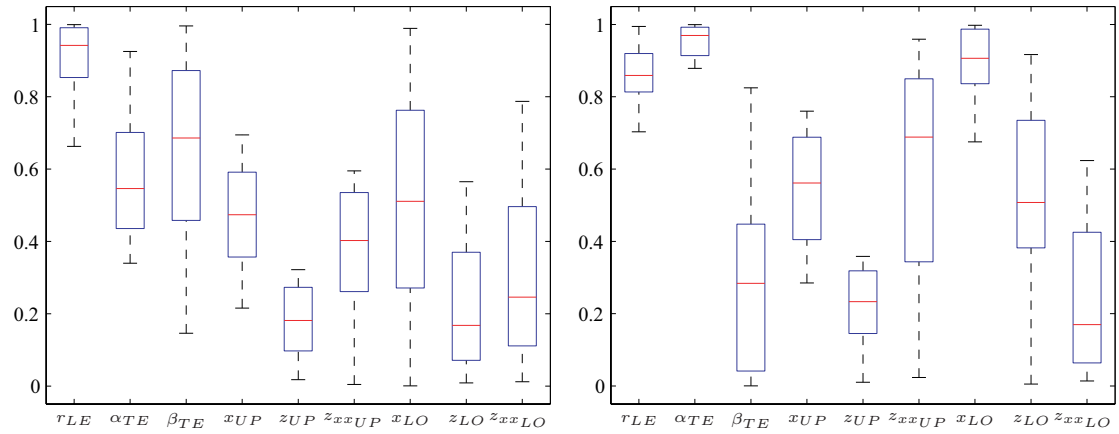
(b) Time-step convergence using the HV metric

Figure 5.18: Final results of the low-fidelity design case-study

the preferred compromise. It provides a 39% and 37% improvement over the reference point design for objectives f_1 and f_2 respectively. The min f_1 design (as expected) yields slighty further improvement in f_1 with a 40% improvement, at the expense of only a 30% improvement in f_2. The min f_2 design instead provides a 62% improvement in f_2 at the expense of only a 32% improvement in f_1. These results clearly demonstrate that the RAE2822 places more emphasis on the f_1 condition, and slight improvements in the f_1

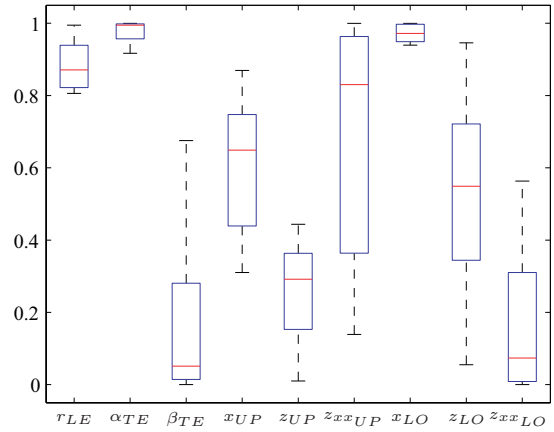
Design	C_d	C_l	C_m	f_1	f_2	d_z
RAE2822	0.008807	0.7938	-0.1215	0.0140	0.0148	–
$\mathbf{z}'$	0.004600	0.7342	-0.0959	0.0085	0.0092	-0.0055
min f_1	0.004833	0.7604	-0.1013	0.0084	0.0103	-0.0045
min f_2	0.003957	0.6445	-0.0747	0.0095	0.0056	-0.0045

Table 5.6: UPMOPSO results of the low-fidelity wave drag optimization study

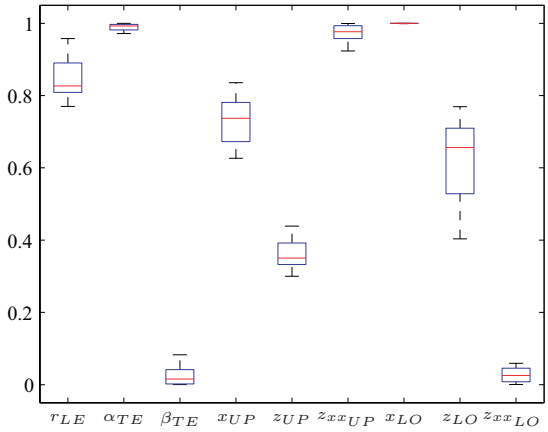


(a) $t = 10$

(b) $t = 20$



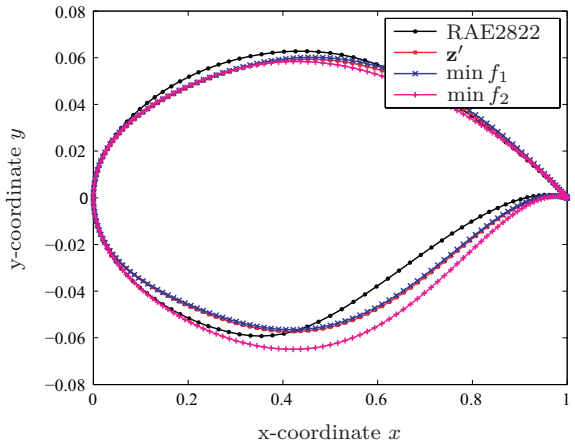
(c) $t = 30$



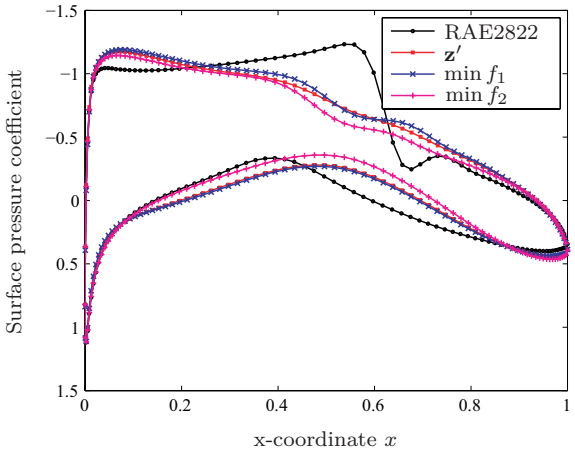
(d) $t = 50$

Figure 5.19: Normalized box-plots of variable ranges at interval time-steps

objective provide drastic changes in the f_2 objective. Despite the *bias* towards f_1, the UPMOPSO algorithm is still able to faithfully reflect the preferred compromise. Another strategy for *visualizing* the convergence of the optimization process is to monitor the variable spread as solutions exploit the preferred region. Figure 5.19 show the variable ranges of the non-dominated solution set at certain time-step intervals. A distinct attraction towards a very specific region of the design space is observed. It is also demonstrated that certain variables are saturated at either the lower or upper boundary, suggesting that the respective ranges should be adjusted. It is evident that the final preferred designs aim to minimize the amount of curvature on the upper surface to reduce flow acceleration which is critical to avoid shock formation. This is further complemented with the increase in leading edge radius, as well as the mild upper thickness. This combination of effects essentially plateaus the upper surface, stabilizing the flow velocity and reducing the adverse pressure gradient. The reduction in pitching moment is due primarily to the reduction in lower surface curvature and trailing edge angle, which is pivotal in reducing the aft



(a) Geometry preferred designs



(b) Surface pressure at $M = 0.78$ and $\alpha = 2^\circ$

Figure 5.20: Comparison of optimized designs and the RAE2822 reference point

loading. The lower thickness is however quite high, and complemented with the delayed lower thickness location, significant lift may still be generated. The profile geometries and surface pressure distributions are shown in Figure 5.20. These plots verify the conclusions derived from Figure 5.19 and Table 5.6. It is shown that in order to optimize f_1 objective, the drag C_d is reduced to a shockless value, and the lift C_l is slightly reduced (a result of the reduction in aft loading). The f_2 objective is achieved by reducing the aft curvature/loading, as reducing the magnitude of C_l (consequently decreasing the lift-to-drag ratio – thereby increasing f_1) is directly proportionate to reducing the C_m magnitude.

5.4 Summary

In this chapter, a comprehensive discussion on the development of the proposed multi-objective particle swarm optimization algorithm has been given. The operation of the

reference point distance metric has been described, and its unique integration into the algorithm has been documented. The convergence behaviour of the swarm was studied, and an insight into controlling the location of the preferred region was provided. To highlight the superiority of the developed algorithm, a series of unconstrained and constrained test suites were considered. The UPMOPSO algorithm was compared with the NSGA-II and R-NSGA-II algorithms in an attempt to provide an unbiased reflection of the performance of UPMOPSO. It was conclusively demonstrated that the UPMOPSO is superior to the NSGA-II variants for all test suites, in both phases of exploration and exploitation. To further demonstrate the convergence behaviour of the algorithm, a low-fidelity aerodynamic design case-study was presented. This case study demonstrates the operation of the algorithm for an aerodynamic design application, and confirms its viability for the high-fidelity design case-studies of Chapter 7.

Chapter 6

Implementing Kriging and Visualization

Aerodynamic design optimization problems benefit from the construction of inexpensive surrogate models that emulate the response of exact functions. This chapter presents a novel development to the field of preference-based optimization. Adaptive Kriging models are incorporated within the swarm framework to efficiently navigate design spaces restricted by a computational budget. The successful integration of these design tools is facilitated through the reference point distance metric, which provides an intuitive criterion to update the Kriging models during the search. Surrogate-based visualization methods are also introduced, which are used to statistically extract information from the design space and confirm the relative influence of both variables and objectives to the preferred interests of the designer. Such visualization techniques may be applied before or after optimization, to facilitate the design or decision making processes.

6.1 Implementing the Kriging Method

An introduction to the surrogate-based optimization architecture was presented in Chapter 4. A frequent occurrence in these studies is the use of the Kriging method, since it inherently considers the confidence interval of the predicted output. This allows the optimizer to exploit the information provided by the Kriging prediction, in determining whether a solution is worthy of precise evaluation. Verifying whether a solution is valid of precise evaluation is the key to implementing the Kriging method. It is also beneficial to maintain the correct direction of the search so no solution is unnecessarily evaluated, such that it may contribute to the global accuracy of the Kriging model yet does not aid the optimization convergence.

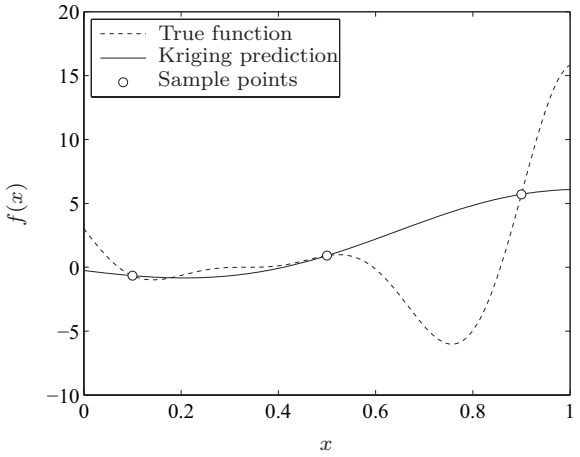


Figure 6.1: One-variable function with Kriging model through three sample points

The operation of the Kriging method was also described in Chapter 4, detailing the procedure to construct and validate a Kriging model which emulates an expensive function landscape. A novel development to the field of preference-based optimization is proposed in this section, describing the implementation of the Kriging method to the preference-based multi-objective swarm algorithm. In this section a review of some traditional update strategies for the Kriging method is presented, before introducing the update strategy developed for this research, which is motivated by the preference-based theme adopted thus far. The aim is to facilitate the synthesis of the design framework, such that the reference point is no longer solely used to dictate the preferred search direction of the swarm, but also provides a guideline in determining the validity of a predicted solution.

6.1.1 Update Strategies

Consider the multi-modal one-variable function, where a Kriging model is constructed based on the three sample points, as shown in Figure 6.1. This section will introduce some popular techniques for updating the surrogate to identify the global minimum. These techniques are critical to the development of the novel reference point screening criterion.

Exploitation

The most intuitive update strategy is to exploit the predicted optimum of the Kriging model. This update strategy will ensure a fast convergence rate, however is very dependent on initial sampling and may fail to find the true global optimum in the presence of a multi-modal function. Referring to Figure 6.1, it is observed that the Kriging method provides a rather crude approximation of the function. By identifying the minimum function value provided by the Kriging model and applying an infill point at that location,

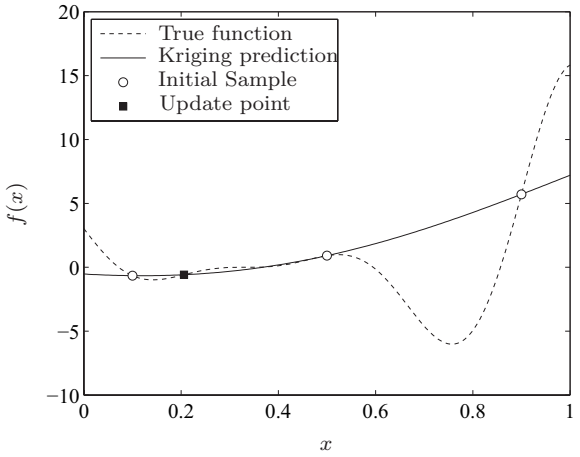


Figure 6.2: Kriging model enhanced by an update point at the identified optimum

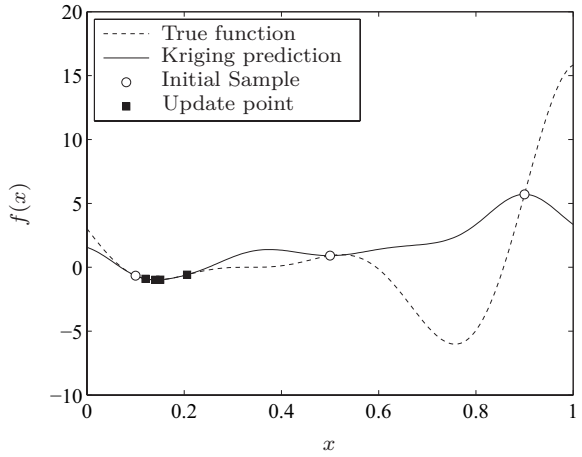
a more accurate approximation in the region of the identified optimum is obtained, as shown in Figure 6.2. It is observed from Figure 6.3(a) that after an additional five update points the Kriging model has identified the region of the local optimum, which has been extensively sampled. The limitations of this strategy are evident here, since this strategy is purely based on exploiting the identified optimum and is not capable of exploring other areas of the design space which may provide further improvement. After eight update points, the Kriging approximation is still trapped within the region of the local optimum, as shown in Figure 6.3(b). Due to the orientation of the original sampling plan, this strategy has not converged to the global minimum, and it is observed that pure exploitation can unnecessarily prolong the search or in fact cause convergence to a false optimum.

Exploration

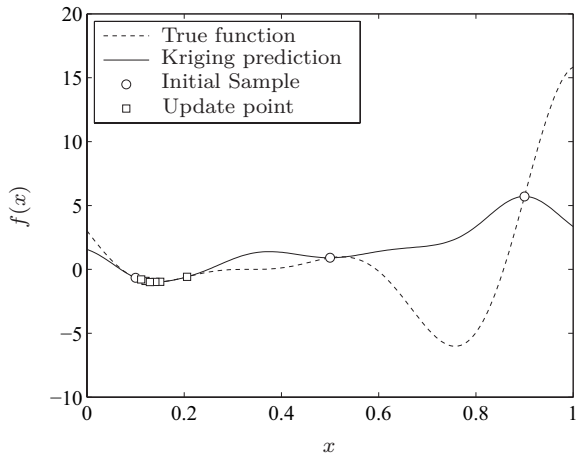
The exploitation strategy is very dependent on the initial sample since it is not capable of exploring other regions of the design space which may offer improvement. Consequently, the full capabilities of the Kriging method are not utilized. The accuracy of the prediction $\hat{y}$ at the unobserved location $\mathbf{x}$ depends on the correlation distance with sample points $\mathbf{X}$. The closer the location of $\mathbf{x}$ to the sample points, the more confidence in the prediction $\hat{y}(\mathbf{x})$. The Gaussian function adopted by the Kriging method thus allows for an estimate to the prediction error. The uncertainty in the prediction is estimated by

$$\hat{s}^2(\mathbf{x}) = \hat{\sigma}^2 \left[1 - \mathbf{r}^T \mathbf{R}^{-1} \mathbf{r} + \frac{(1 - \mathbf{1}^T \mathbf{R}^{-1} \mathbf{r})^2}{\mathbf{1}^T \mathbf{R}^{-1} \mathbf{1}} \right]. \tag{6.1}$$

On examination of Eq. (6.1), it is observed that if the error $\hat{s}(\mathbf{x})$ is calculated at a sample point $\mathbf{x}_i$, then $\mathbf{x} \subset \mathbf{X}$ and thus $\mathbf{r}$ is a column of $\mathbf{R}$. Hence the expression $\mathbf{R}^{-1}\mathbf{r}$ is the i-th unit vector leading to $\mathbf{1}^T \mathbf{R}^{-1} \mathbf{r} = 1$ and $\mathbf{r}^T \mathbf{R}^{-1} \mathbf{r} = 1$. Substituting these proofs into



(a) Additional five update points



(b) Additional eight update points

Figure 6.3: Kriging model with additional sample points based on previous optima

Eq. (6.1) yields the error $\hat{s}(\mathbf{x}) = 0$. This follows the characteristic of any interpolation method, whereby a previously sampled point is precisely recorded and stored and therefore there is no prediction error.

A possible update strategy as a direct result of the error prediction is to infill points at locations which have maximum error. This update strategy is potentially immune to the multi-modality of any function landscape. Figure 6.4 revisits the example of the one-variable function where four additional update points which have been selected based on pure exploration. The accompanying plot which features the error distribution is shown in Figure 6.5. It is observed that such a process will eventually identify the global minimum of the function, however is analogous to simply increasing the sample plan to ensure constant error across the entire design space. Furthermore, since this strategy is based on pure exploration, there is no precursor as to when the search can confidently cease.

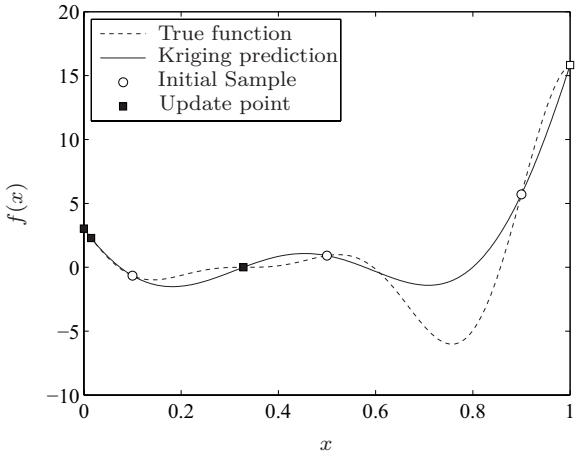


Figure 6.4: Kriging model enhanced by four update points based on pure exploration

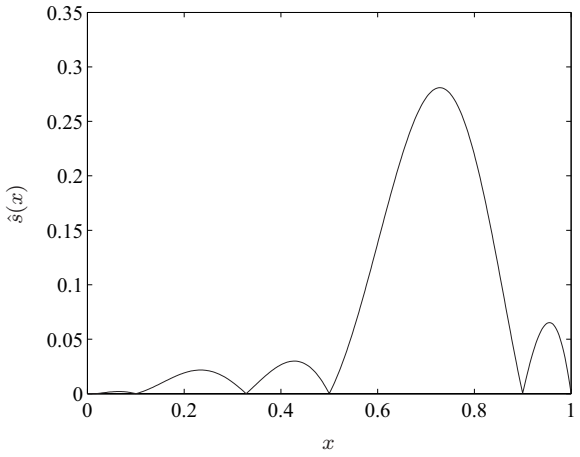


Figure 6.5: Kriging prediction error based on the approximation shown in Figure 6.4

Balanced Exploration and Exploitation

Thus far, update strategies which are either based on pure exploitation and exploration have been presented. Pure exploitation results in an accelerated convergence rate, but can not necessarily guarantee convergence to the true global minimum. On the other hand, pure exploration can potentially identify the global minimum of any function, yet it is highly inefficient for an optimization process. The logical conclusion is to combine the strengths of both update strategies in an effort to eliminate the deficiencies. A simple and effective method [see Forrester *et al.*, 2008], of balancing exploration of the error $\hat{s}(\mathbf{x})$ and exploitation of the prediction $\hat{y}(\mathbf{x})$ is to minimize the lower-confidence bound:

$$\hat{y}(\mathbf{x})_{lb} = \hat{y}(\mathbf{x}) - \omega \cdot \hat{s}(\mathbf{x}), \tag{6.2}$$

where ω is a constant that controls the bias between exploration and exploitation. As $\omega \to 0$, the result is pure exploration and similarly if $\omega \to \infty$, the effect of $\hat{y}(\mathbf{x})$ becomes

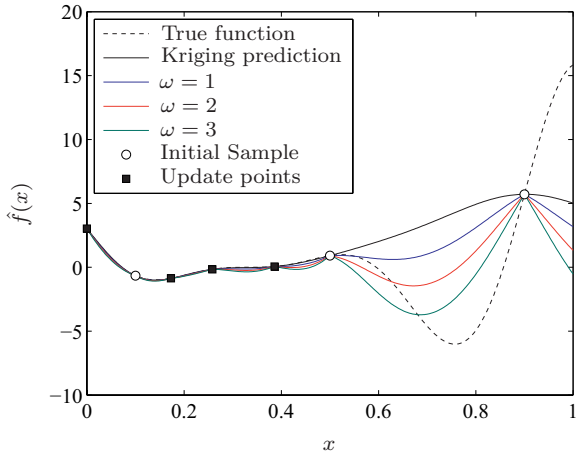


Figure 6.6: The lower bound strategy for balancing exploration and exploitation

negligible and is equivalent to pure exploration. The consensus of the lower-confidence bound strategy is to direct the infill points in promising but less explored regions of the design space. Emmerich *et al.* [2006] demonstrated this criterion in the context of optimization of multi-modal functions, where significant improvement in results over exploitation methods was observed. A reasonable choice for the user-defined parameter is $\omega = 2$, which leads to a 97% confidence probability that $\hat{y}(\mathbf{x})_{lb}$ is the lower-confidence bound of $\hat{y}(\mathbf{x})$. Figure 6.6 shows the lower bound for varying ω after four additional update points. It is shown that this method is quite capable of identifying promising areas of the design space, without the added expense of sampling in areas which have large error values.

The choice of the extra parameter provides some ambiguity in the use of the lower bound strategy, as Figure 6.6 highlights that the strategy could potentially arrive at the global minimum, yet the most feasible value is ultimately dependent on the function landscape. Other update strategies have been developed which are free from choosing such a parameter. The probability of improvement, or more specifically the *expected improvement* is one such technique which has been popularized for surrogate-based engineering design applications. This criterion calculates the improvement which is expected from the predicted solution at location $\mathbf{x}$, with reference to the minimum solution $Y_{\min}$ found so far. Figure 6.8 illustrates the predicted expected improvement over the design landscape based on the approximation shown in Figure 6.7. The expected improvement of any given solution can be obtained as,

$$E[I(x)] = \begin{cases} Y_{\min} - y(\mathbf{x})\Phi\left(\frac{y_{\min}-y(\mathbf{x})}{\hat{s}(\mathbf{x})}\right) + s\phi\left(\frac{Y_{\min}-y(\mathbf{x})}{\hat{s}(\mathbf{x})}\right) & \text{if } s > 0 \\ 0 & \text{if } s = 0 \end{cases} \tag{6.3}$$

where Φ and ϕ are the cumulative distribution function and probability density function respectively. Rather than searching for the minimum lower bound value, one could ef-

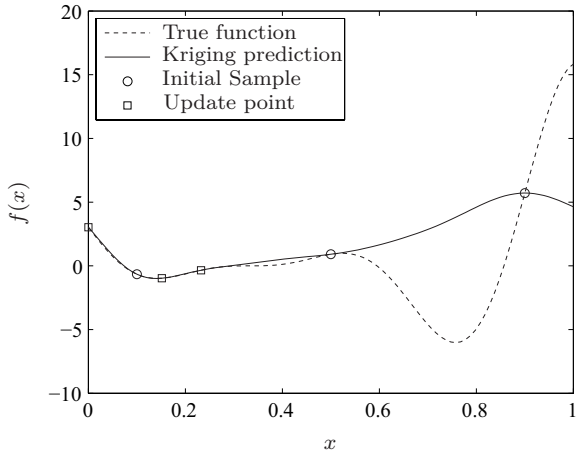


Figure 6.7: Kriging model enhanced by three update points based on expected improvement

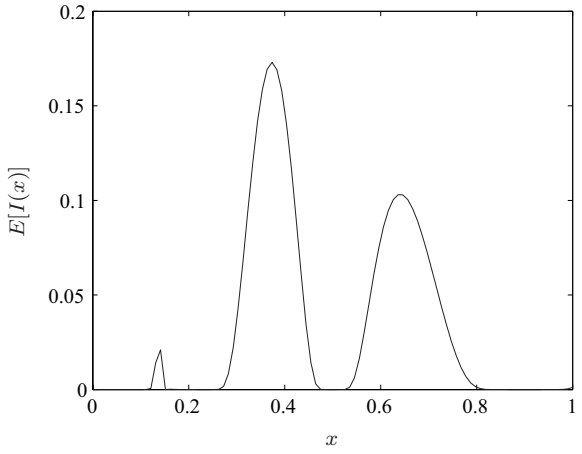


Figure 6.8: Predicted expected improvement based on the approximation shown in Figure 6.7

fectively balance exploration and exploitation by maximizing the expected improvement. Whether a sampled solution exploits a local basin of attraction or samples a relatively unknown location is irrelevant and is only dependent on where the improvement is predicted. This strategy is by far the most intuitive and complete strategy to finding the global minimum of a function and has been included in a number of aerodynamic design frameworks [Emmerich *et al.*, 2006; Jeong *et al.*, 2005a; Jones, 1998; Ong *et al.*, 2004].

6.1.2 Reference Point Screening Criterion

The developed reference point screening criterion draws on the concept of individual-based adaptive control [see Jin, 2005] and is motivated by the lower bound update strategy. Kriging fitness predictions $\hat{f}()$ are used to screen each candidate particle in the swarm after the population update (or likewise after mutation). The Kriging model estimates

Algorithm 7 Reference point screening strategy

```
for each particle x_i do
  PREDICT f̂(x_i)
  PREDICT d̂_z(x_i)
  if d̂_z(x_i) < z'(t) then
    EVALUATE f(x_i)
    EVALUATE d_z(x_i)
    RETRAIN Kriging models [X; x_i]
    UPDATE archive Q(t + 1)
  end if
end for
```

the lower-confidence bound (lb) of the objective array as,

$$\{\hat{f}_1(\mathbf{x}), \ldots, \hat{f}_m(\mathbf{x})\} = [\{\hat{y}_1(\mathbf{x}) - \omega \cdot \hat{s}_1(\mathbf{x})\}, \ldots, \{\hat{y}_m(\mathbf{x}) - \omega \cdot \hat{s}_m(\mathbf{x})\}]. \tag{6.4}$$

The factor ω is specified as 2 to provide a 97% probability that $\hat{f}_i(\mathbf{x})$ is the lower-bound value of $\hat{y}_i(\mathbf{x})$. An approximation to the reference point distance is thus computed as,

$$\hat{d}_z(\mathbf{x}) = \max_{i=1:m} \left\{ \left(\hat{f}_i(\mathbf{x}) - \bar{z}_i \right) \right\}. \tag{6.5}$$

The approximation to the reference point distance provides an estimate to the improvement that is expected from the solution. At time t, the most preferred archive member is recorded as $\mathbf{z}'(t)$. Particle candidates are subsequently flagged for precise evaluation or rejection as described in Algorithm 7. Stratified sampling using an LHS methodology is used to construct a global Kriging approximation [$\mathbf{X}$, $\mathbf{Y}$] with least number of sample points (N_K). The non-dominated subset of $\mathbf{Y}$ is used to construct the non-dominated archive as outlined in Section 5.2.1. This ensures that candidates for global leadership have been precisely evaluated (or with negligible prediction error) and, therefore, offer no false guidance to other particles. Mathematically the candidate global leaders at time t are a subset of the non-dominated archive $\mathbf{Q}(t)$ which is a subset of the Kriging dataset $\mathbf{X}$, such that

$$\mathbf{X}_g(t) \in \mathbf{Q}(t) \in \mathbf{X}_{1\ldots N_K}. \tag{6.6}$$

Based on the success of the Kriging prediction, a candidate particle may be subsequently flagged for precise evaluation. This evidently requires retraining of the Kriging models, and can subsequently lead to reconstruction of the global archive.

The lower bound screening criterion ensures particles will be attracted towards the areas of the design space which provide the greatest resemblance to $\bar{\mathbf{z}}$ and the direction of the search will remain consistent. A similar screening strategy adopting the expected

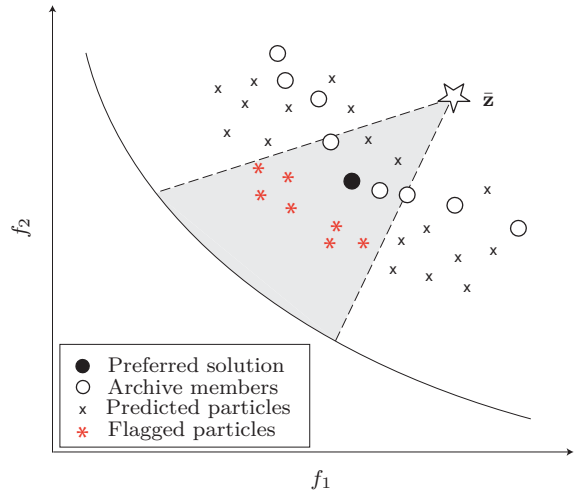
improvement criterion was tested, yet failed to provide similar results. There are a number of motives for adopting the lower confidence bound strategy (at least in conjunction with the reference point method) over the widely more popular expected improvement approach. Firstly, the expected improvement criterion leads to no logical strategy for determining whether a solution is deemed feasible for exact evaluation, which contradicts the adaptive control strategy (see Section 4.1.2). For example, the candidate particle with highest expected improvement may intuitively be selected, but there is ambiguity in specifying a *threshold* value or a minimum value of expected improvement. This would likely result in performing an unnecessary, or likewise an inadequate, number of evaluations during the search. Secondly, the success of the expected improvement criterion is greatly dependent on the value of $Y_{\min}$. This guides the particles to explore areas which expected improvement in either objective rather than the preferred compromise. This could be alleviated through a strategy which predicts the expected improvement in the reference point distance rather than any specific objective, but is beyond the scope of this research.

Fortunately by adopting the lower confidence bound strategy, many of the earlier reported ambiguities are resolved. As the search initially commences and the prediction accuracy of the Kriging model(s) is low, there will be a large percentage of the swarm that is flagged for precise evaluation. Subsequently, as the particles begin to identify the preferred region and the prediction accuracy of the surrogate model(s) gradually increases, the screening criterion becomes increasingly difficult to satisfy, thereby reducing the number of flagged particles at each time-step. This is further illustrated in Figure 6.9 which shows instances of exploration and exploitation.

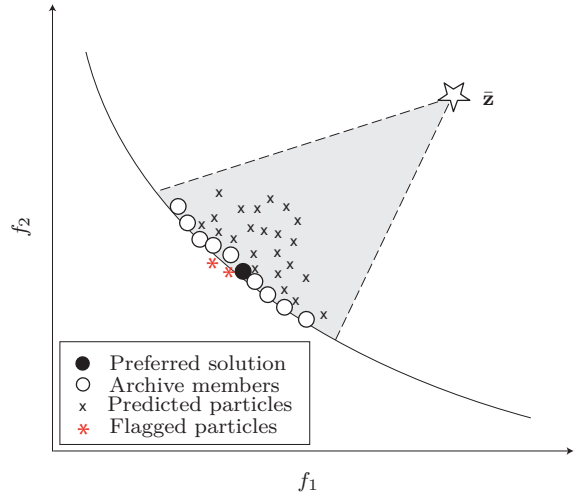
In Figure 6.9(a) at time t_1 it is shown that the preferred region is not well defined. Particles are initially predicted using the lower bound strategy which overestimates their success. The flagged particles (however likely these particles really do offer the level of improvement that is predicted) ensure the Kriging models are updated to offer greater improvement in the preferred region. In Figure 6.9(b) at time t_2 it is shown that the preferred region is now well defined, such that the screening criterion is difficult to satisfy. This ensures the preferred region is exploited, without the risk of performing unnecessary evaluations in other unexplored regions of the design space.

6.2 Kriging UPMOPSO Algorithm

The proposed Kriging UPMOPSO (or the KUPMOPSO) algorithm amalgamates the search strategy of the UPMOPSO algorithm (see Section 5.2.5) with the developed Kriging



(a) Illustration of initial stages of search at time t_1



(b) Illustration of final stages of search at time t_2

Figure 6.9: Flagging successful particles based on the reference point screening procedure

reference point screening criterion for performing high-fidelity aerodynamic design optimization studies. This algorithm presents a novel addition to the field of preference-based optimization. The pseudo-code of the KUPMOPSO algorithm is presented in Algorithm 8. The stopping criterion is intuitively based on the maximum number of precise function evaluations $f_{\max}$ as dictated by the allowable computational budget. An extensive validation procedure was performed for the UPMOPSO algorithm in Section 5.3. The search effort of the KUPMOPSO algorithm remains essentially the same with the exception that the number of *precise* evaluations are significantly reduced due to the inclusion of the Kriging models. Presented in this section are a series of experimental problems. The aim here is not to assess the searching prowess of the method (which has been conclusively demonstrated), but rather the effect of the screening criterion.

Algorithm 8 The Kriging UPMOPSO (KUPMOPSO) algorithm

OBTAIN user-defined preferences, $\mathbf{z}$ and δ
CONSTRUCT Kriging models $[\mathbf{X}, \mathbf{Y}]$ of N_K samples; $t = 0$
CONSTRUCT archive $\mathbf{Q}(t+1)$
RECORD most preferred solution $\mathbf{z}'(t+1)$
INITIALIZE swarm of size $[N, n]$
PREDICT fitness and lower bound distance metric
ASSIGN personal best
$t = 1$
repeat
  SELECT global leaders
  UPDATE particle velocity
  CONSTRICT velocity
  UPDATE particle position
  APPLY boundary conditions
  PREDICT fitness and lower bound distance metric
  SCREEN particles for precise evaluation
  EVALUATE successful particles
  RETRAIN Kriging models
  UPDATE archive $\mathbf{Q}(t+1)$
  RECORD most preferred solution $\mathbf{z}'(t+1)$
  UPDATE personal best
  if MUTATION triggered **then**
    MUTATE archive members
  end if
  $t = t + 1$
until $f_{\max}$

6.2.1 Schaffer Test Function

The Schaffer mathematical test function is a bi-objective problem of the form:

$$\min f_1(\mathbf{x}) = \frac{2}{\gamma} \cdot \left(\sum_{i=1}^{d} \mathbf{x}_i^2 \right)^{\gamma/2} \tag{6.7}$$

$$\min f_2(\mathbf{x}) = \frac{2}{\gamma} \cdot \left[\sum_{i=1}^{d} (1 - \mathbf{x}_i)^2 \right]^{\gamma/2} \tag{6.8}$$

where the design space range is $\mathbf{x} \in [0, 10]^d$, the number of dimensions $d = 10$ and $\gamma = 2$. The curvature of the Pareto front is scalable by the parameter γ. The convex Pareto front follows the equation,

$$y_2 = \left(1 - y_1^{1/\gamma}\right)^{\gamma} \qquad y_1 \in [0, 1]. \tag{6.9}$$

The Schaffer test function is not characterized as deceptive; though a random initial population tends to be reasonably far from the global Pareto front, which generally prolongs the

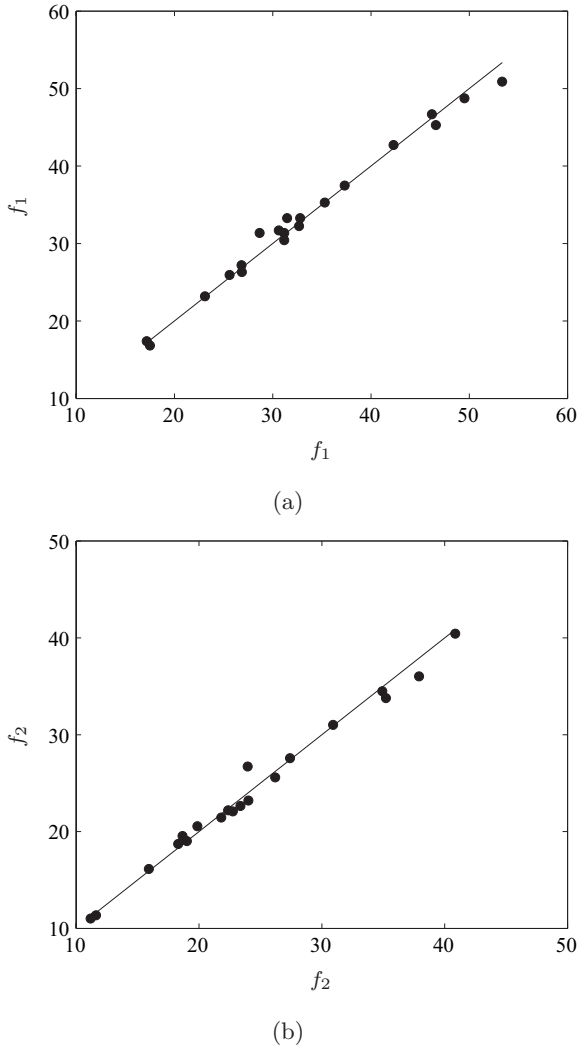


Figure 6.10: Cross-validation for the constructed Kriging models of the Schaffer test function

explorative phase of the search. By focusing exclusively on the preferred region and constructing a Kriging model for each objective, the time spent in exploration and hence the number of function evaluations can be significantly reduced. Simulations are performed with a population of $N = 100$ individuals using the benchmark NSGA-II algorithm, the UPMOPSO algorithm and the KUPMOPSO algorithm. For the latter, a Kriging model is constructed for each objective based on an LHS of $N_K = 20$ design points. Figure 6.10 illustrates the cross-validation error plot of the constructed Kriging models.

The reference point is selected as $\bar{z} = [0.2,\ 0.2]$ with a solution spread of $\delta = 0.015$. With no broadly applicable stopping criterion, all algorithms stopped after a specified number of function evaluations. Results for the NSGA-II and the UPMOPSO algorithms are recorded after 1000 and 5000 function evaluations. For the KUPMOPSO algorithm, results are recorded after 150 function evaluations.

Algorithm	**Evaluations**	$\mathbf{z}'$	$\max(d_z) - \mathbf{z}'$
NSGA-II	1000	7.6201 ± 1.8600	–
NSGA-II	5000	0.5371 ± 0.2823	–
UPMOPSO	1000	0.1742 ± 0.0274	–
UPMOPSO	5000	0.0553 ± 0.0009	0.0159 ± 0.0013
KUPMOPSO	150	0.0521 ± 0.0003	0.0152 ± 0.0002

Table 6.1: Optimization results for the convex Schaffer test function

Statistical results (mean $\pm$ standard deviation) of ten independent simulations are shown in Table 6.1. The most preferred ($\mathbf{z}'$) is indicative of the closeness of the non-dominated solution set to the global Pareto front. The KUPMOPSO consistently provides a more accurate solution set over the standard UPMOPSO algorithm and NSGA-II algorithm at approximately 2% of the computational cost (the number of function evaluations refers to the sample points plus the additional update points). Results for the variance ($\max(d_z) - \mathbf{z}'$) of the non-dominated solution set are also recorded; this provides a measure of the uniformity in solution spread. Since the NSGA-II algorithm explores the full extent of the Pareto front, the variance is not recorded. Larger values of the variance suggest a non-uniform scattering of solutions, whilst values that approach δ indicate the desired solution spread is attained.

It is clearly observed from Figure 6.11 that the KUPMOPSO algorithm identifies a set of Pareto-optimal solutions in the immediate vicinity of the reference point. Also shown are selected sample points used to update the Kriging models. The proficiency of the reference point pre-screening criterion is evident from the sample point attraction towards the preferred region. This establishes a distinct search direction which is beneficial

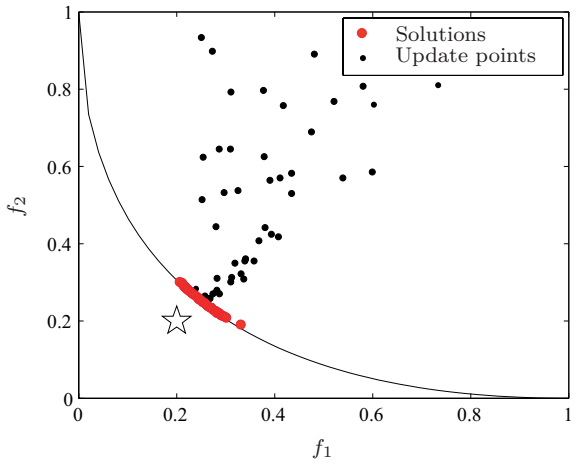


Figure 6.11: KUPMOPSO on Schaffer function showing update points in the immediate vicinity

to screen prospective swarm particles to determine whether they are feasible for precise evaluation. It is also observed that only few precisely evaluated solutions reside on the Pareto front. This information is sufficient to predict a uniform spread of solutions on the Pareto front with negligible error.

6.2.2 Design of a Helical Compression Spring

Presented here is a multi-objective constrained problem which was first proposed by Tudose and Jucan [2007]. A subsequent study, using surrogate models, is described by Forrester *et al.* [2008]. A helical compression spring is to be designed to work over a stroke of $h = 50$mm with a corresponding load variation between $F_{\min} = 40$N and $F_{\max} = 500$N. ASTM A229/SAE J315 oil tempered wire is used with modulus of elasticity $E = 2.06 \times 10^5$ MPa, density $\rho = 7.87 \times 10^{-6}$ kg/mm^3 and rigidity modulus $G = 0.78 \times 10^5$ MPa. There are two conflicting objectives, 1) to minimize the mass and 2) to maximize the fatigue life of the spring. Two constraints ensure that the spring does not fail in shear and in buckling respectively. There are three design variables, whose ranges are shown in Table 6.2. The first is the wire diameter d. The second is the index i, defined as the ratio of the mean helix diameter (measured from the center of the wire) and the diameter d. The final variable is the maximum load intercoil distance coefficient k_Δ, which is the ratio of the distance between adjacent coils of the fully loaded spring and the diameter d.

Shown in Figure 6.12 are the results of the evaluation of a 250,000 point full-factorial sampling plan. Only a marginal percentage (i.e. 66,139) of the designs have proven to be feasible, where they do not violate the two resistance constraints and they make geometrical sense (i.e. their variable triplets generate sensible springs). For a conventional (exhaustive) search, this would suggest a high number of unnecessary (or at least avoidable) evaluations are performed.

As the following results will indicate, the KUPMOPSO algorithm performs efficiently for this problem, in spite of the very large infeasible subdomain of the design space. Simulations are performed with a population of 100 individuals using the NSGA-II algorithm, the UPMOPSO algorithm and the KUPMOPSO algorithm. For the latter, a Kriging

Variable	**Lower bound**	**Upper bound**
d (mm)	0.5	7
i	4	16
k_Δ	0.1	1.1

Table 6.2: Ranges of variables for the design of a helical compression spring

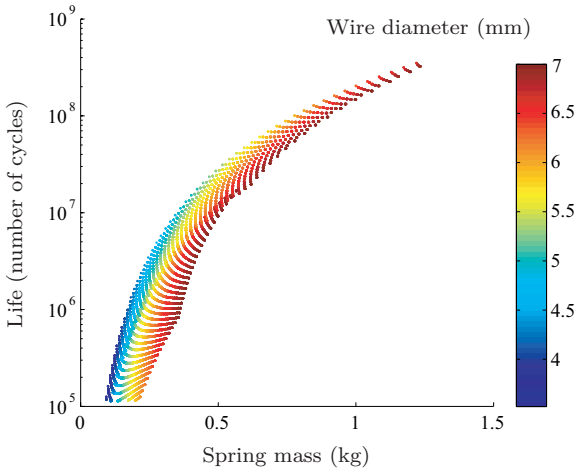


Figure 6.12: Full factorial plan and the corresponding wire diameters of the feasible designs

model is constructed for the objective and constraint functions based on a LHS sample of 20 design points, of which only 7 designs satisfy the constraints (Note: designs which yield geometrically non-sensible springs are omitted). The reference point is specified as the ideal target design of 0.1 kg mass, and a life of 10^{10} cycles. The solution spread is specified as $\delta = 0.015$. For the NSGA-II and UPMOPSO algorithms, results are recorded after 2000 function evaluations. For the KUPMOPSO algorithm, results are recorded after 50 function evaluations.

Statistical results (mean $\pm$ standard deviation) of 10 independent simulations are shown in Table 6.3. A representative simulation is shown in Fig. 6.13 compared with the Pareto-front trend estimated from the full-factorial search of Figure 6.12. The KUPMOPSO algorithm is observed to be far more proficient in comparison to the other algorithms, because it consistently obtains more accurate (and uniform) results at a fraction of the computational cost. Furthermore, the percentage of feasible solutions (i.e. sensible designs with no record of constraint violation) is sufficiently greater. Although the objectives and constraints are inexpensive to compute for this specific problem, if each experiment were a computer simulation or destructive test, the potential time and cost savings would be significant.

Algorithm	**Evaluations**	**% Feasible**	$\mathbf{z}'$	$\max(d_z) - \mathbf{z}'$
NSGA-II	2000	40.2 ± 0.5	0.2925 ± 0.0014	–
UPMOPSO	2000	41.7 ± 0.7	0.2902 ± 0.0027	0.0152 ± 0.0023
KUPMOPSO	50	68.7 ± 0.2	0.2877 ± 0.0005	0.0151 ± 0.0002

Table 6.3: Optimization results for the helical spring design problem

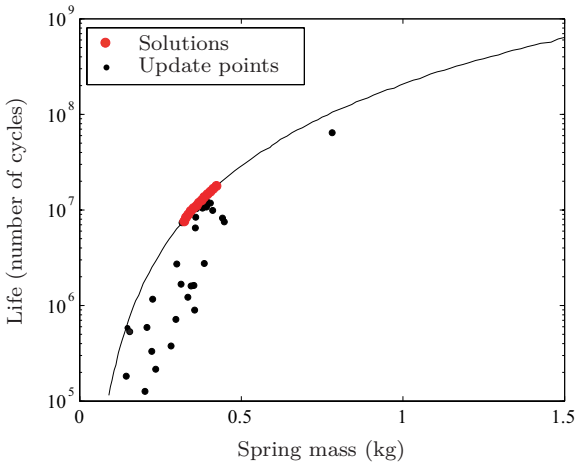


Figure 6.13: KUPMOPSO on helical spring design problem showing feasible sample points

6.2.3 Low-fidelity Design: Case-study Revisited

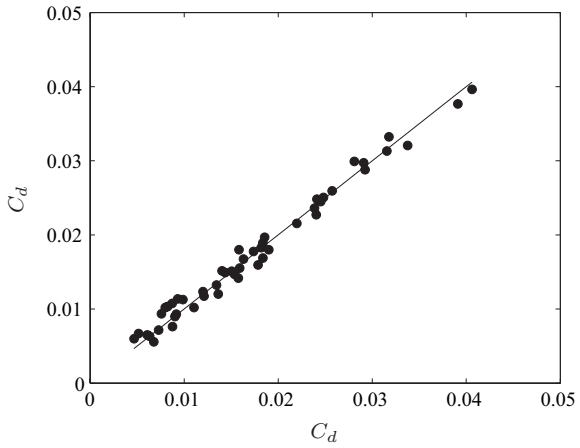
The low-fidelity case-study (see Section 5.3.5) is revisited in an attempt to demonstrate the dramatic increase in computational efficiency using the KUPMOPSO algorithm, whilst still maintaining the same search proficiency as the UPMOPSO counterpart. In this case, it is beneficial to construct the Kriging models to replicate the aerodynamic coefficients rather than the objective functions, as to avoid square and reciprocal square terms. A stratified sample based on an LHS methodology is used to construct the global Kriging approximations of $N_K = 50$ samples. These global approximations $\mathbf{Y}_{1,\ldots,3}$ are initially trained using the cross-validation procedure, where the error plots are illustrated in Figure 6.14. These error values are considered acceptable for the optimization study. The objective functions are thus computed as:

$$\hat{f}_1(\mathbf{x}) = \left(\hat{y}_1(\mathbf{x}) - \omega \cdot \hat{s}_1(\mathbf{x})\right) / \left(\hat{y}_2(\mathbf{x}) - \omega \cdot \hat{s}_2(\mathbf{x})\right)^2 , \tag{6.10}$$

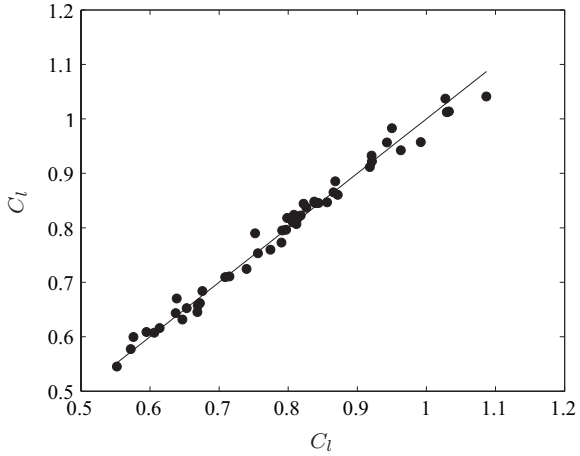
$$\hat{f}_2(\mathbf{x}) = \left(\hat{y}_3(\mathbf{x}) - \omega \cdot \hat{s}_3(\mathbf{x})\right)^2 . \tag{6.11}$$

A swarm population of $N = 100$ particles is initialized and flown for a maximum computational budget of 200 precise evaluations. The Kriging datasets are restricted to 150 samples during the optimization process, in an attempt to localize the prediction accuracy of the models to preferred regions only, and to reduce training time. A further 150 precise updates were performed over $t = 228$ time-steps until the computational budget was breached. Figure 6.15 illustrates the error margins of the constructed Kriging models, captured at $t = 0$ and $t = 228$. There is a distinct reduction in the mean and variance of the error for all models, owing mostly to the localized focus of the preferred region.

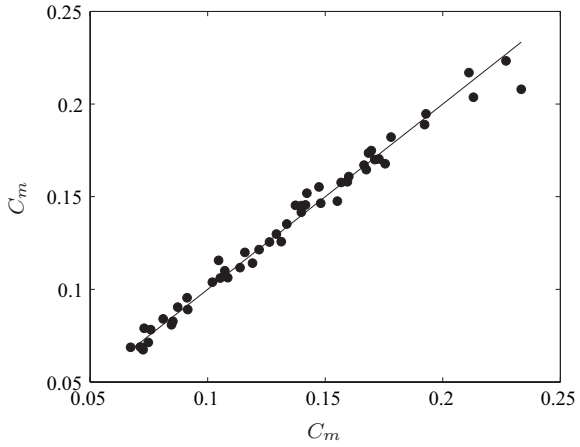
The performance of KUPMOPSO is illustrated in Figure 6.16. It is shown from Figure 6.16(b) that the largest number of update points is recorded during the initial phases



(a) $CV(\mathbf{Y}_1) = 3.20\%$

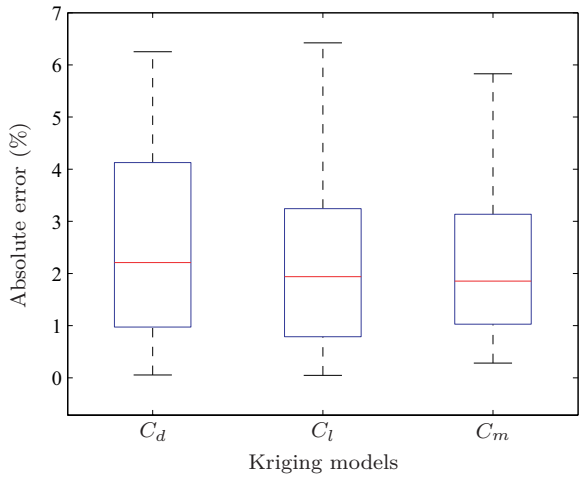


(b) $CV(\mathbf{Y}_2) = 3.01\%$

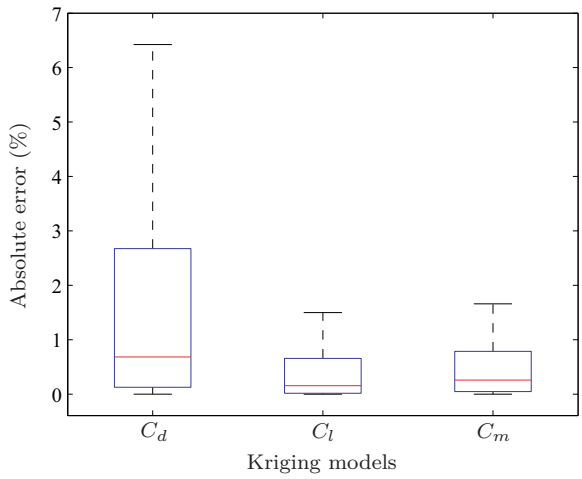


(c) $CV(\mathbf{Y}_3) = 3.44\%$

Figure 6.14: Cross-validation curves for the constructed Kriging models of the low-fidelity design case study based on the aerodynamic coefficients.



(a) $N_K = 50$; $t = 0$

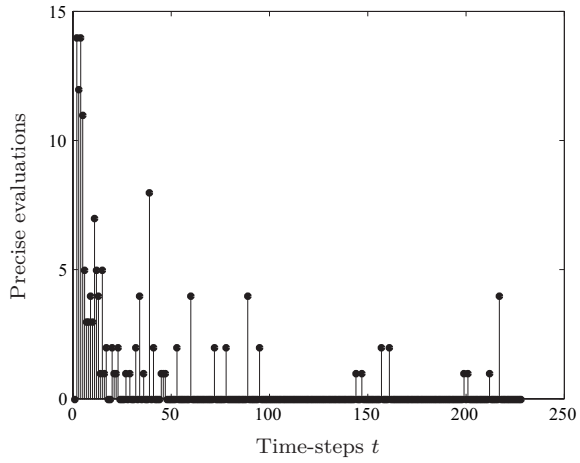


(b) $N_K = 150$; $t = 228$

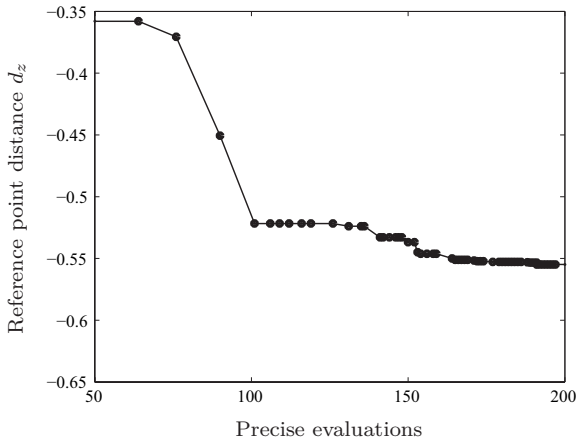
Figure 6.15: Illustration of error margins via box-plots for the constructed Kriging models

of the search. As the preferred region becomes populated and $\hat{\mathbf{s}} \to 0$, the search enters into an exploitation phase and the number of update points steadily reduces. Figure 6.16(a) features the progress of the most preferred design as the number of precise evaluations increase. The reference point criterion is shown to be proficient in filtering out poorer solutions during exploration, since only 50 update evaluations are required to reach within 15% of the final identified design and a further 50 evaluations to reach within 5%. Furthermore, no needless evaluations as a result of the lower-bound prediction are performed during the exploitation phase. This conclusion is further complemented by Figure 6.17(a), as a distinct attraction to the preferred region is clearly visible.

The final set of non-dominated solutions identified is shown in Figure 6.17(b), superimposed over the set obtained by the standard UPMOPSO algorithm. It is evident that



(a) Number of evaluations per time-step



(b) Progress of the most preferred solution $\mathbf{z}'$

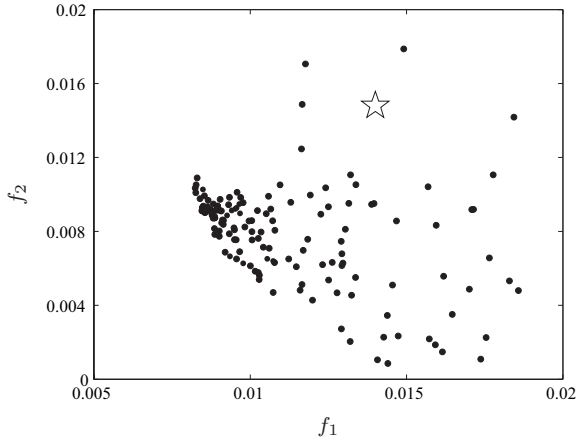
Figure 6.16: Performance of the KUPMOPSO algorithm for the low-fidelity design case-study

the set of non-dominated solutions is not as uniformly spread as that identified by the UPMOPSO algorithm - a consequence of implementing the Kriging component - which is an inevitable deficiency of the KUPMOPSO algorithm[1]. However the search process remains fairly consistent, where a similar spread of optimal designs has been identified despite the dramatic reduction in computational effort.

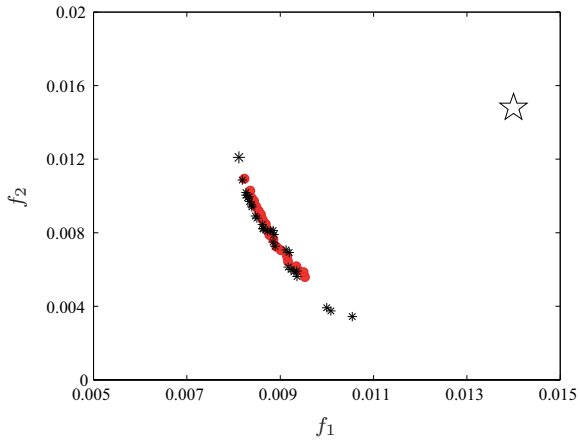
6.3 Visualization Strategies

In Section 4.4 a series of visualization techniques were described. It was observed that such techniques could potentially offer meaningful information on the design space or objective space during all phases of optimization. As has been a recurring theme in this thesis the

[1]Precise evaluations are performed based on the success of the Kriging *prediction* - it can never be assured that a solution will provide the same performance as obtained from the Kriging prediction



(a) Illustration of most recent update points



(b) Solutions of UPMOPSO (•) and KUPMOPSO (*)

Figure 6.17: Precise evaluations performed and the resulting non-dominated designs

low-fidelity case study is revisited in an attempt to demonstrate the practical application of some of the visualization techniques that have been documented. A pre-optimization and variable screening phase is introduced, to measure the level of importance of each PARSEC variable on the objective landscapes. Despite this problem having multiple objectives (or quantities of interest), it is shown how the relative importance of each variable can be measured by relating its sensitivity to the deviation from the preferred compromise. A post-optimization decision making process is also introduced. SOM charts are used to visualize the identified trade-offs and characterize them according to their similarity to the reference point compromise.

6.3.1 Pre-optimization and Variable Screening

The global Kriging models constructed in Section 6.2.3 are considered sufficient in order to obtain sufficient confidence in the results of the subsequent visualization analyses. Whilst

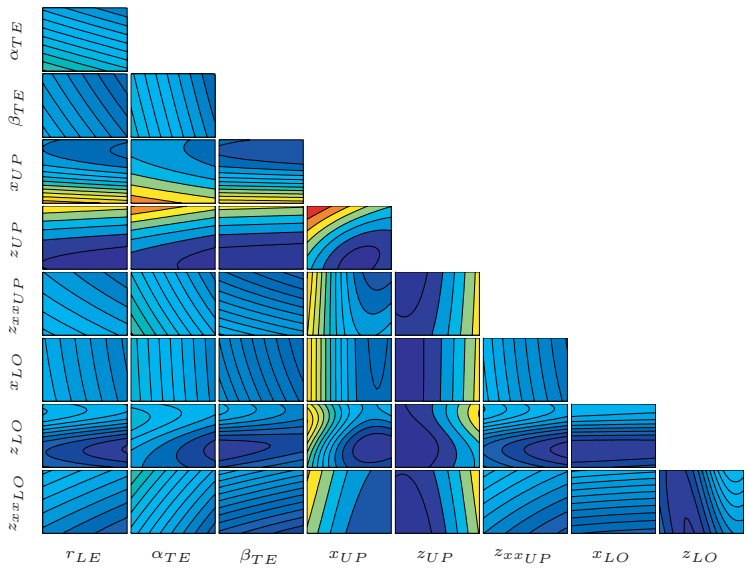


Figure 6.18: Matrix of two-dimensional contour plots (coloured by drag coefficient)

a larger sampling plan is essential to obtain an accurate global correlation, it is only of interest to quantify the elementary effect of each variable to the objective landscapes. The baseline value is intuitively selected as the reference point RAE2822 airfoil, which is mapped using the PARSEC method via a least-squares method. Figure 6.18 maps the variable relationship with the drag coefficient C_d utilizing the methodology discussed in Section 4.4.1.

Figure 6.18 provides some insight into the non-linearity of the objective landscape, and it is observed that the variables relating to thickness and surface curvature have the greatest effect. However within a multi-objective context, investigating the elementary effect of variables through contour slices does not provide a complete overview of the *optimization* landscape. For example, the trailing edge variables (α_{TE} and β_{TE}) do not seem to have a significant impact on the drag coefficient, but may have a large influence on the pitching moment. To avoid this deficiency, variable screening is conducted using the Morris algorithm (see Section 4.4.2). Table 6.4 consolidates the results of the analysis and based on the mean and standard deviation, conclusions are made whether a variable is influential and/or interacting with other variables.

The screening study immediately provides two important observations. Firstly it is shown that the trailing edge wedge angle β_{TE} has a negligible influence and is essentially inactive. This is most likely caused by the restricted boundaries of the variable range, or its inactivity could also be due to the fidelity of the potential flow method (e.g. the inclusion of the boundary layer could activate the influence of β_{TE}). This variable could safely

	Drag C_d		**Lift** C_l		**Moment** C_m	
Variables	**Influential**	**Interactive**	**Influential**	**Interactive**	**Influential**	**Interactive**
r_{LE}			✓			
α_{TE}		✓	✓		✓	✓
β_{TE}						
x_{UP}	✓	✓	✓	✓	✓	✓
z_{UP}	✓	✓	✓	✓	✓	✓
$z_{xx_{UP}}$	✓		✓			✓
x_{LO}						✓
z_{LO}	✓	✓	✓	✓	✓	✓
$z_{xx_{LO}}$		✓	✓	✓	✓	✓

Table 6.4: Results of the variable screening study for the low-fidelity case-study

be omitted from the design space as it is not likely to provide any influence on the final designs. Other inactive variables are identified as r_{LE} and x_{LO} and are observed to have only isolated influence on the aerodynamic quantities. The second important observation is that most variables do not influence considerably the aerodynamic coefficients but they are mostly all non-linear and/or involved in interactions. Only the leading-edge radius r_{LE} is shown to be highly active for computing the lift coefficient, but has a negligible effect on other quantities.

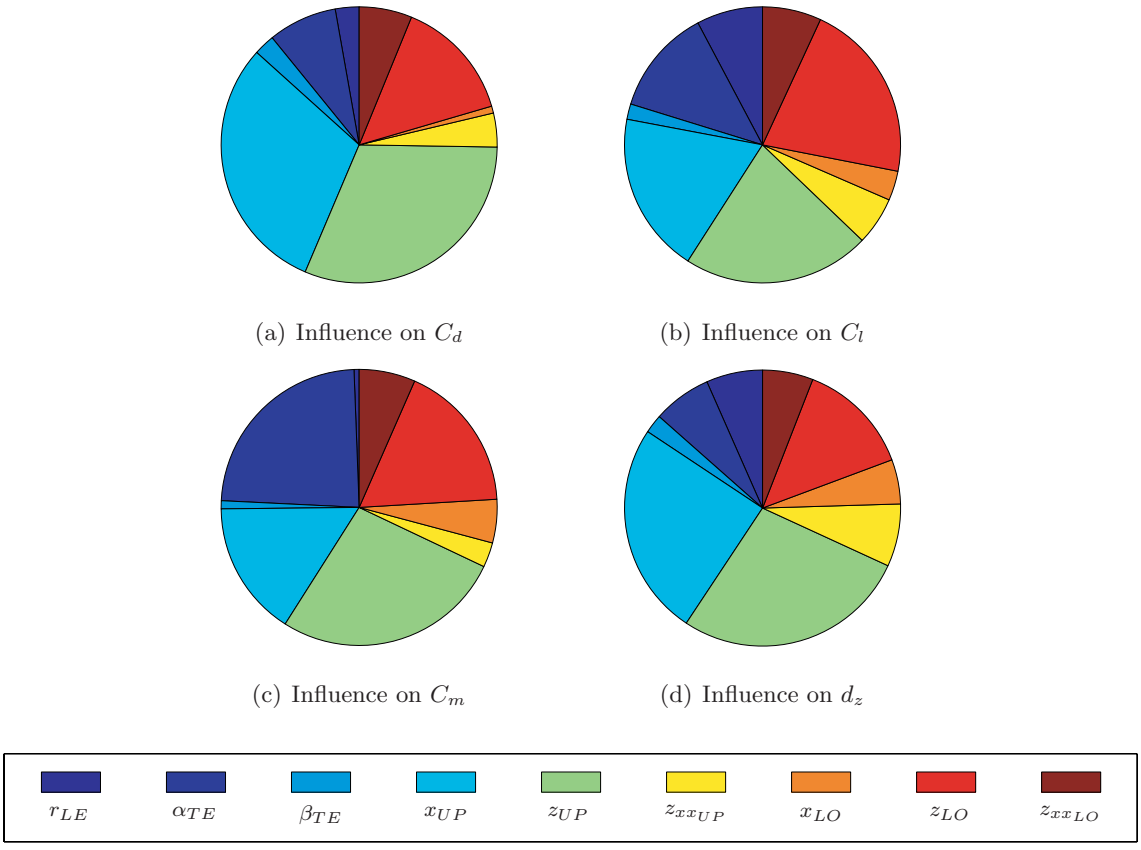
While this strategy provides the designer a clear insight into the variable influence on the aerodynamic quantities, once again it offers no insight on the optimization landscape. Figure 6.19 alternatively features the results of the extended Morris algorithm [Saltelli *et al.*, 2001], where variable elementary effects are now screened using the quantity μ^*. It is evident that the upper and lower thickness is of significant importance for all aerodynamic quantities. While the lower thickness location x_{LO} does not have much influence on the lift and drag coefficients, it is shown to have a greater effect on the pitching moment, as this variable to some degree will control the aft camber. It is observed that Figures 6.19(a)-6.19(d) faithfully reflect the conclusions obtained from Table 6.4, with the exception that the charts are simpler to interpret.

Of particular significance is Figure 6.19(d) which examines the variable influence on the deviation from the reference point compromise. The variable influence on d_z is case-specific and entirely dependent on the reference point chosen for the proposed optimization study. Since the value of d_z is a means of ranking the success of a multi-objective solution as one single scalar, variables may be ranked by influence, which is otherwise not possible when considering a multi-objective array. It is therefore argued that while Figures 6.19(a)-6.19(c) measure variable influence on the isolated objective functions, Figure 6.19(d) mea-

sures the variable influence on the actual *optimization* landscape. Preliminary conclusions to the priority weighting of the objectives to the reference point compromise can also be made. For example, it is interesting to note that r_{LE} has a negligible influence on all quantities except on the lift coefficient, however its influence on d_z is fairly evident, suggesting that the lift coefficient (i.e. f_1) drives the design. It is observed that Figure 6.19(d) more closely resembles the charts of Figures 6.19(a) and 6.19(b) which further concludes that the design driver is the lift-to-drag ratio, a conclusion which compliments the findings of previous studies. The reader should be fully aware that this conclusion is case-dependent and based on the selection of the reference point, i.e. with variations to the reference values the variable influence on d_z will not be consistent. It is also important to note that while Table 6.4 suggests inactivity of the variables r_{LE}, β_{TE} and x_{LO} - only the trailing edge wedge angle β_{TE} is observed to be negligible.

6.3.2 Post-optimization and trade-off visualization

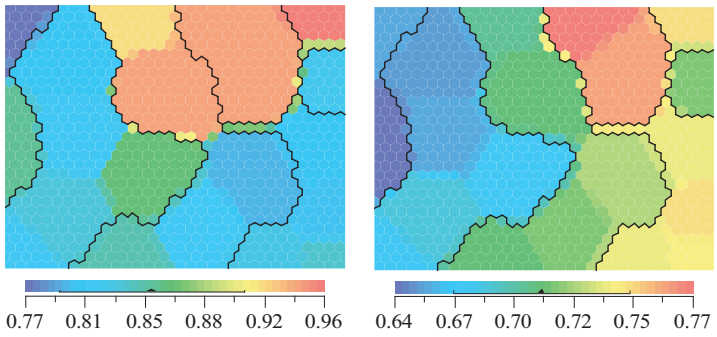
The pre-optimization phase focuses on setting up the optimization problem and providing a statistical measure on the variable influence. The designer is generally unaware of



(a) Influence on C_d (b) Influence on C_l

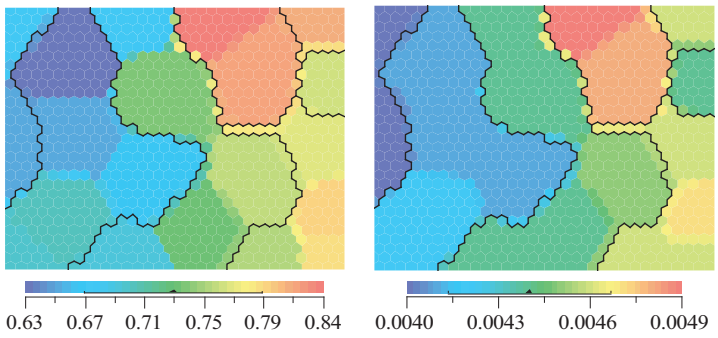
(c) Influence on C_m (d) Influence on d_z

Figure 6.19: Results of the extended Morris screening study for the low-fidelity case-study



(a) Coloured by r_{LE} (b) Coloured by C_l

Figure 6.20: Visualization of the influence of variable r_{LE} on the lift coefficient C_l

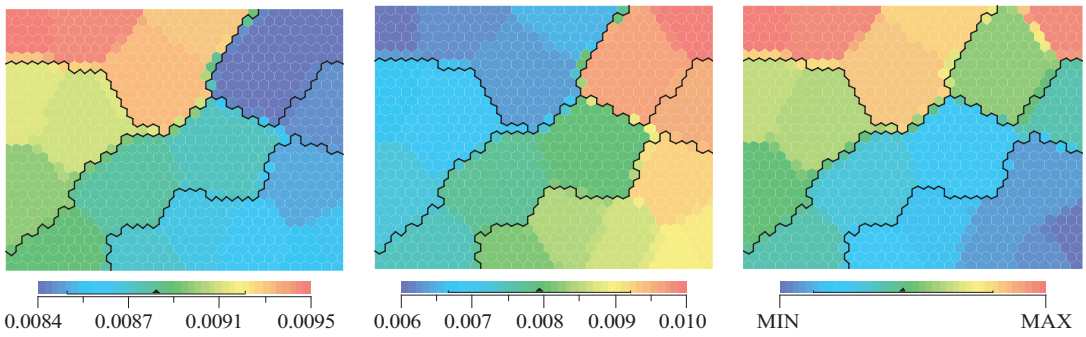


(a) Coloured by x_{UP} (b) Coloured by C_d

Figure 6.21: Visualization of the influence of variable x_{UP} on the drag coefficient C_d

the best performing areas of the design space, and thus pre-optimization is performed (reluctantly) over the entire design space. Once the optimization has been performed, the area of the design space for visualization is significantly reduced. Furthermore, the activity and influence of the variables within the best performing areas of the design space can be properly assessed. Figures 6.20 and 6.21 illustrate selected charts of a SOM analysis (see Section 4.4.3) based on a stratified sample of the localized preferred region. The data is arranged via seven clusters based on increasing values of the aerodynamic coefficients. Figure 6.20 illustrates the activity of the leading edge radius r_{LE} on the lift coefficient, while Figure 6.21 illustrates the activity of the upper thickness location x_{UP} on the drag coefficient. Both variables were concluded to be significant to the respective coefficients. The qualitative data provided by the SOM chart illustrates the variable activity and provides a comprehensible insight into the variable influence.

It was shown in the previous section how the reference point facilitated the screening of variables to measure their influence on the optimization landscape. The reference point additionally provides a feasible means of selecting the most appropriate solution to aid the



(a) Coloured by f_1 (b) Coloured by f_2 (c) Coloured by d_z

Figure 6.22: Visualizing the compromise between the design objectives via SOM representation

decision making process. For example, identified designs may be ranked according to how well they represent the reference point compromise. This is achieved through the use of SOM. To visualize the compromise between the aerodynamic coefficients and the reference point a SOM is trained to evaluate the relationship between the inputs C_d, C_l and C_m to the output d_z. A user-defined cluster separator is integrated to partition designs based on increasing values of d_z.

Each neuron in this case refers to a potential solution of which the designer may select the one that best suits the intended application. The charts are organized via five clusters, which facilitate the interpretation of the SOM analysis. Observing the chart in Figure 6.22(c) it is clearly demonstrated that minimum values of the reference point distance metric d_z are obtained by reducing the f_1 objective at the expense of higher f_2 values. This suggests that the preferred compromises places a greater emphasis on the lift-to-drag objective, a conclusion which has already been obtained in earlier analyses. This is further complemented by the fact that the d_z chart more closely resembles the f_1 chart which suggests that the deviation from the preferred compromise is directly proportionate to the f_1 objective. Utilizing these charts, it is shown that the post-optimization and decision making process is simplified, and the addition of the d_z chart allows for the scalar ranking of identified solutions. While this particular case-study only involves two objectives, which does not particularly warrant the use of multi-dimensional data mining techniques, the use of SOM for problems of three or more objectives is highly advantageous.

6.4 Summary

This chapter describes the integration of surrogate modelling to the preference-based optimization framework. For high-fidelity problems where the objective function is expensive, Kriging models may be constructed to alleviate the computational burden. Popular screen-

ing strategies for locally updating the Kriging models during the optimization process are first described. Thereafter, the development of a novel screening criterion is documented, which utilizes the information provided by the reference point to screen candidates. This is a simple criterion based on the lower-confidence bound strategy, and is shown to be proficient in identifying solutions which are expected to provide improvement within the preferred region of the design space. To demonstrate the operation of the surrgoate-based framework (or the KUPMOPSO algorithm) a number of analytical problems are first attempted, before revisiting the low-fidelity aerodynamic design case-study introduced in the previous chapter. The search effort is shown to remain consistent as the UPMOPSO algorithm, with the added advantage that less precise evaluations are performed. The chapter concludes with a demonstration of the visualization strategies that were introduced in Chapter 4. Using the low-fidelity aerodynamic design case-study, it is shown how the results of the screening analyses are comprehended using the reference point distance metric. This allows candidate designs of a multi-objective problem to be assessed as a single scalar. Variables which directly drive the optimization process (rather than the objective landscape) are easily identified. A methodology to visualize resulting trade-off solutions using self-organizing maps is also presented. In this case, the reference point distance metric is used to characterize idenfitifed designs, and relate them to the preferred interests of the designer to facilitate the decision making process.

Chapter 7

Case-studies and Results

The design tools discussed in the previous chapters are synthesized to develop an efficient framework for aerodynamic design. A wide spectrum of applications is presented in this chapter that adopt various design philosophies. For each application, the optimization framework is combined with a suitable shape parameterization tool and robust computational solver based on the required fidelity. For selected problems, data mining aids are applied to visualize the design landscape and investigate the compromise between design objectives. The preferred interests of the designer are reflected through selection of the reference point. This is an intuitive way of articulating designer preferences, since a reference point may be ideally based on an existing or target design. On the basis of these case-studies, the practical efficiency of the preference-based optimization framework is demonstrated and its proficiency in navigating complex optimization landscapes is highlighted.

7.1 Multi-mission Airfoil Shape Optimization

This section describes the attempt at utilizing the design framework for a typical subsonic airfoil shape optimization scenario (previously published in Carrese *et al.* [2011a]). The aim for this design case-study is to broaden the multi-mission capabilities of subsonic Unmanned Aerial Vehicles (UAV).

Despite advantages at one flight condition, a good design is one that exhibits optimal aerodynamic performance over a range of flight conditions and mission segments. Traditional concepts of developing platforms for single mission requirements have resulted in a large number of UAV with difficulties in operation and support [Khurana *et al.*, 2008a]. Future mission requirements have confirmed that the single mission design concept is neither operationally nor financially feasible. Alternatively, a multi-mission platform is regarded

as a viable design concept to address the issues with the current UAV fleets [Khurana *et al.*, 2008a]. Present UAV platforms are plagued with performance restrictions and operations outside the intended design envelope are not permissible. For example long endurance UAV are optimized for slow speed operations. Performing high *g-force* manoeuvres with these platforms would result in suboptimal performance. This application focuses on the design of UAV airfoils for multi-mission capabilities.

Problem Formulation

The formulated objectives cover mission segments that are typical of multi-mission UAV requirements. The minimization of the drag coefficient (C_d) is first considered, at a fixed lift coefficient (C_l), Reynolds number (Re) and Mach number (M) such that,

$$f_1 = \min C_d \text{ at } C_l = 0.5,\ Re = 4 \times 10^6,\ M = 0.3. \tag{7.1}$$

Providing a reduced drag during cruise is generally achieved at the expense of a highly aft cambered airfoil section which results in excessive pitching moments. An objective is thus formulated to minimize the zero-lift pitching moment coefficient, which is desirable for stability and control. The incidence angle which corresponds to zero-lift (α_0) is determined. The pitching moment at this condition (C_{m_0}) is recorded such that,

$$f_2 = \min(C_{m_0})^2 \text{ at } C_l = 0,\ Re = 4 \times 10^6,\ M = 0.3. \tag{7.2}$$

UAV are required to perform manoeuvres without the risk of stall. The final objective considers maximizing the highest possible lift coefficient ($C_{l_{\max}}$) before stall occurs, where

$$f_3 = \min 1/100 \cdot C_{l_{\max}}^2 \text{ at } Re = 4 \times 10^6,\ M = 0.3. \tag{7.3}$$

The objectives described here are considered sufficient to address the multi-mission design philosophy at the preliminary level.

Shape Parameterization

Airfoil shapes are parameterized using the PARSEC method (see Section 2.2.3 – Figure 2.2). In this study, blunt trailing edges sections are not considered and $\Delta z_{TE} = 0$. A thickness constraint of 10% chord is applied by appropriately setting the ranges of the thickness variables x_{UP} and x_{LO}. The PARSEC ranges are as shown in Table 7.1. The suggestions of Khurana and Winarto [2010] has been followed, in an attempt to organize the design space such that it is reflective of the class of subsonic airfoils desired. For example, a significantly large range is assigned for both the leading edge radius r_{LE} (to control the forward suction peak and lift generation) and the upper surface curvature $z_{xx_{UP}}$ (to

Variable	Lower bound	Upper bound
r_{LE}	0.006	0.015
α_{TE}	$-20°$	4.5°
β_{TE}	4.5°	15°
x_{UP}	0.2	0.5
z_{UP}	0.05	0.12
$z_{xx_{UP}}$	-0.95	-0.4
x_{LO}	0.2	0.5
z_{LO}	-0.075	-0.5
$z_{xx_{LO}}$	0.05	0.8
z_{TE}	-0.02	0.02

Table 7.1: PARSEC parameter ranges for the multi-mission airfoil shape optimization study

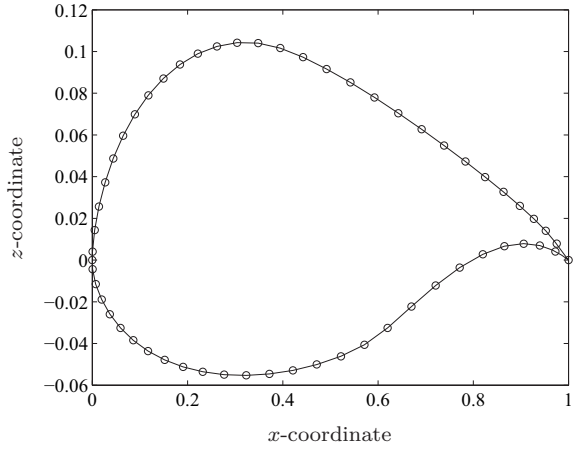
maintain a favourable pressure gradient over the upper surface). The trailing edge angle α_{TE} (representing the aft camber) and the lower surface curvature $z_{xx_{LO}}$ are also assigned large ranges to provide flexibility in balancing the pressure loading.

Flow Solver

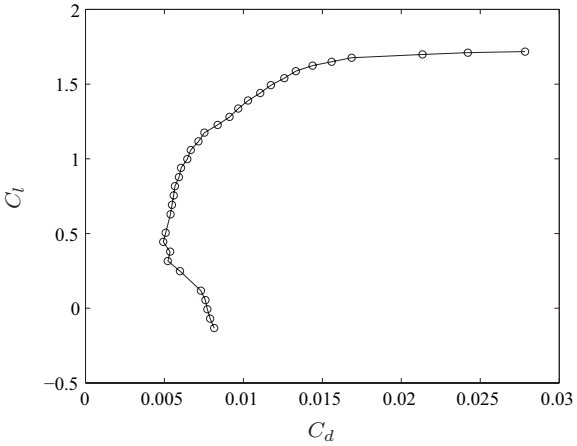
Multi-mission UAV platforms operate within the linear subsonic regime, therefore panel methods are considered to provide a sufficient level of accuracy. The iterative viscous-inviscid airfoil software XFOIL is selected (see Section 2.3.3). XFOIL provides relatively accurate results for subsonic airfoil analysis rapidly using potential and integral boundary-layer theory. Integrated within XFOIL is an iterative design loop, which automatically identifies the aerodynamic properties based on specific operating constraints (e.g. design α or design C_l). This eliminates the need to solve performance constraints explicitly. XFOIL provides an estimate of $C_{l_{\max}}$ which is obtained by monitoring the amount of upper-surface separation of the boundary layer. This separation of the boundary layer induces a de-cambering effect which reduces the value of C_l at post-maximum values.

The Reference Point Design

The reference point selected for this case-study is the NLF0416. It is considered a benchmark profile for low-speed applications since there is experimental data available and it is applicable to general aviation purposes [see Somers, 1981]. The UPMOPSO framework will attempt to improve on the performance characteristics of this airfoil, whilst maintaining a similar level of compromise between the design objectives. Figure 7.1 features the geometrical and aerodynamic properties of the reference point NLF0416 airfoil.



(a) NLF0416 geometrical profile



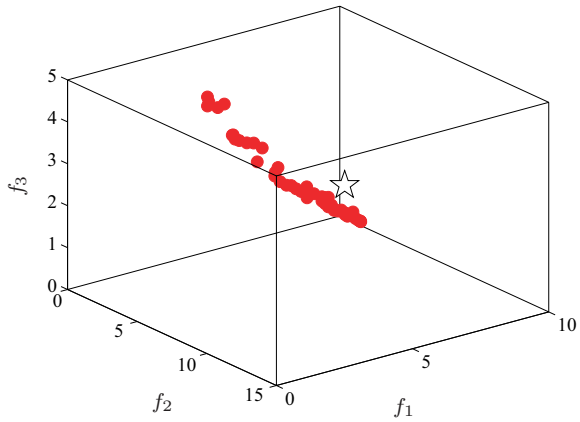
(b) NLF0416 drag polar at $Re = 4 \times 10^6$, $M = 0.3$

Figure 7.1: Geometric and aerodynamic profile of the reference point NLF0416 airfoil

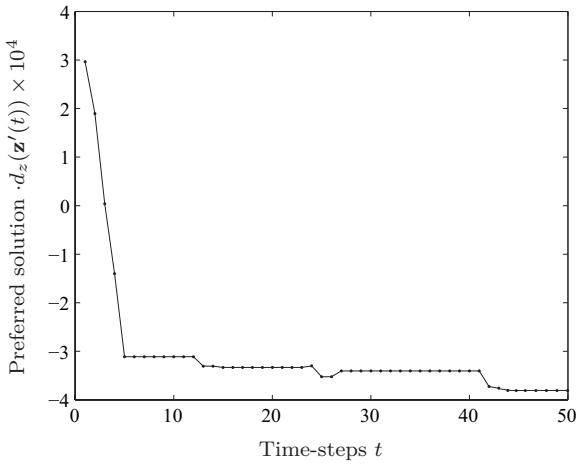
The NLF0416 is a 16% thick airfoil which has a relatively high upper-surface curvature to maintain a favourable pressure gradient for laminar flow control, whilst providing increased maximum lift values for manoeuvring due to the highly cambered aft section. Figure 7.1(b) shows the aerodynamic drag polar computed using XFOIL. It is observed that within the design regime of $C_l = 0.5$, there is a noticable reduction in the C_d. The gradient of the drag polar is fairly mild up to values of $C_l \approx 1.25$, thereafter significant increases in C_d are observed with only slight increments in C_l.

Optimization Results

XFOIL provides results in seconds thereby negating the use of Kriging models to enhance computational efficiency. A swarm population of $N = 100$ particles is flown for $t_{\max} = 50$ time-steps. At each time-step, the swarm is partitioned across four parallel processors, yielding an average computational time per time-step of approximately six minutes (Intel



(a) Identified non-dominated solutions ($\mathbf{f} \times 10^3$)



(b) Progress of the most preferred solution

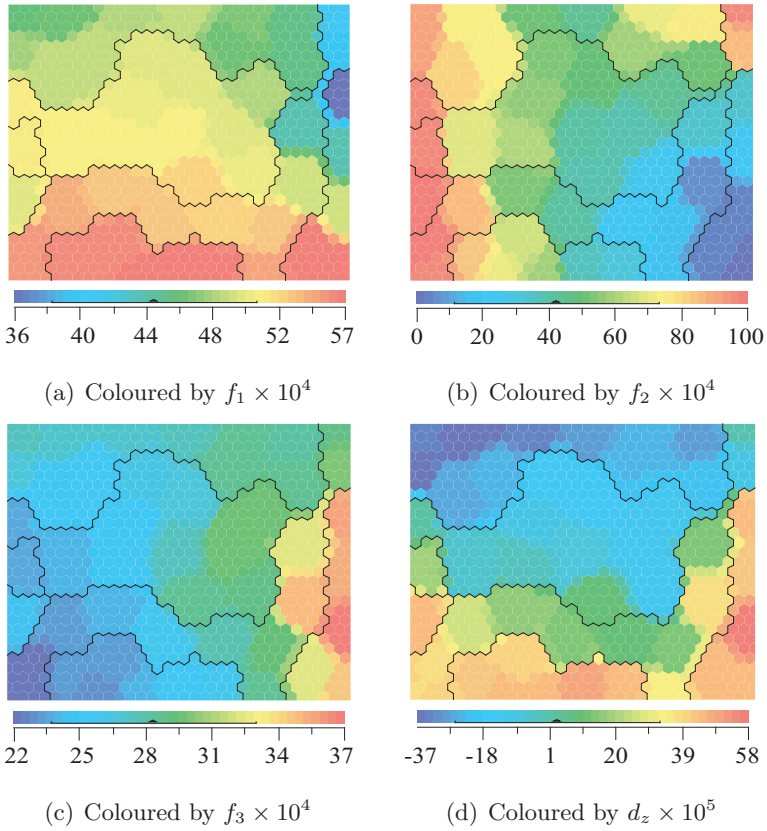
Figure 7.2: Optimization results of the multi-mission airfoil design case-study

Core2 Duo CPU 3.16 GHz). The reference point is obtained by computing the objective values for the NLF0416 using XFOIL and a solution spread of $\delta = 1 \times 10^{-3}$ is specified.

Figure 7.2(a) features the final set of non-dominated solutions identified. The UP-MOPSO algorithm was successful in obtaining a partial subset of non-dominated solutions in the vicinity of the preferred region as dictated by the reference point. The progress of the solution with closest resemblance to the reference point (or the most preferred solution $\mathbf{z}'(t)$) is featured in Figure 7.2(b). It is immediately observed that the most improvement in the solution occurs during the explorative phase of the search, since only 15 time-steps are required to reach within 11% of the final converged design.

Trade-off Visualization

Figure 7.3 features the trade-off visualization analysis conducted using SOMine, where SOM charts are organized by eight clusters according to ascending values of d_z. Solutions



(a) Coloured by $f_1 \times 10^4$ (b) Coloured by $f_2 \times 10^4$

(c) Coloured by $f_3 \times 10^4$ (d) Coloured by $d_z \times 10^5$

Figure 7.3: Trade-off visualization for the multi-mission airfoil design case-study

which yield negative values of d_z indicate improvement over each reference value. Solutions with positive d_z values do not surpass each reference value, but provide significant improvement in at least one other objective. From Figure 7.3(d) it is possible to visualize the trade-off between design objectives. Figure 7.3(d) does not resemble any of the function charts indicating that all objectives hold a similar priority. However, on observing the *hot and cold* sections of the charts it is evident that the most preferred designs offer low-to-mid values for objectives f_1 and f_3 and mid-to-high values for f_2. This clearly suggests that the f_2 objective is of least importance, based on the compromising characteristics of the NLF0416. If the designer were inclined towards a specific objective, then designs which place more emphasis on the respective objective could be considered.

Final Designs

A description of the identified set of non-dominated solutions is provided from the box-plot shown in Figure 7.4. It is immediately observed (even without reference to the geometrical profiles) that the identified airfoils are representative of subsonic (or natural-laminar-flow) airfoils. It is shown that all designs have a relatively high leading edge radius, which is necessary to control upper surface acceleration to avoid flow separation and assist in gen-

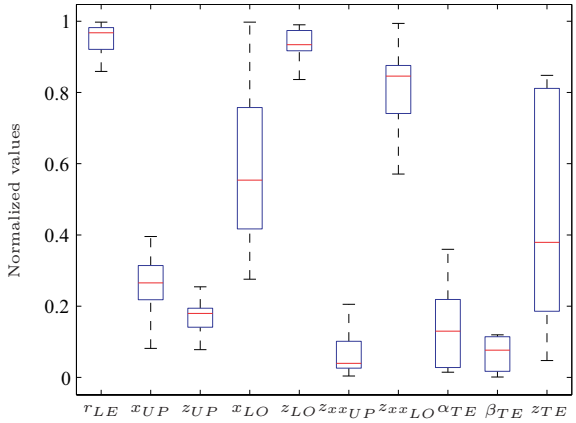


Figure 7.4: Normalized box-plots of variable ranges of identified solutions

erating greater lift. This is accompanied with a very high curvature of the upper surface geometry, which is necessary to prolong laminar flow. The camber is fairly symmetrical, owing to the high curvature of the lower surface. It is however observed that the lower surface thickness location appears fairly downstream - generating the characteristic lower surface *s-shape* essential for controlling the pressure load distribution and hence the pitching moment. The trailing edge is fairly asymmetrical, providing significant trailing edge camber to generate the design lift coefficient.

To demonstrate the performance of the identified solution set, the designs which exhibit the highest performance in either objective are documented. The preferred airfoil design is representative of the solution which provides the greatest resemblance to the NLF0416 reference point compromise. Table 7.2 consolidates the results of the solution spread. The solutions which provide minimum values in either objective clearly do not provide poor performance in other objectives (relative to the performance of the NLF0416) due to the compromising influence of the reference point. It is however confirmed that the f_2 objective is of least importance since large fluctuations are observed across the solution set. This suggests that the f_2 objective does not overly conflict with the other design objectives and could potentially be treated as a constraint in subsequent studies.

Design	f_1	f_2	f_3
NLF0416	0.00516	0.00982	0.00308
Preferred	0.00472	0.00717	0.270
min f_1	0.00407	0.00982	0.00289
min f_2	0.00621	0.00008	0.00326
min f_3	0.00555	0.01018	0.00221

Table 7.2: Results of the multi-mission airfoil optimization study

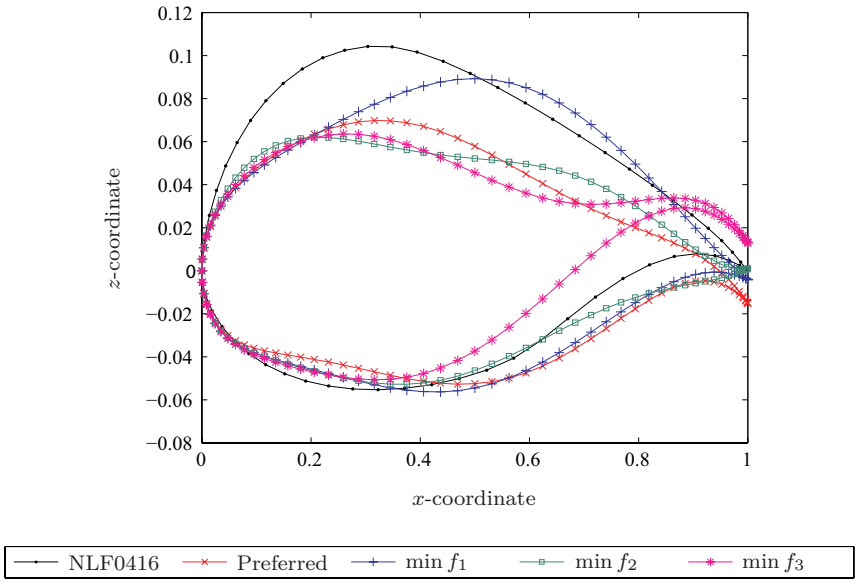
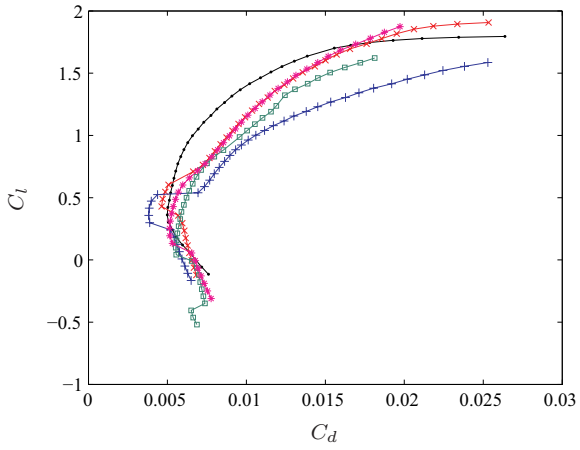


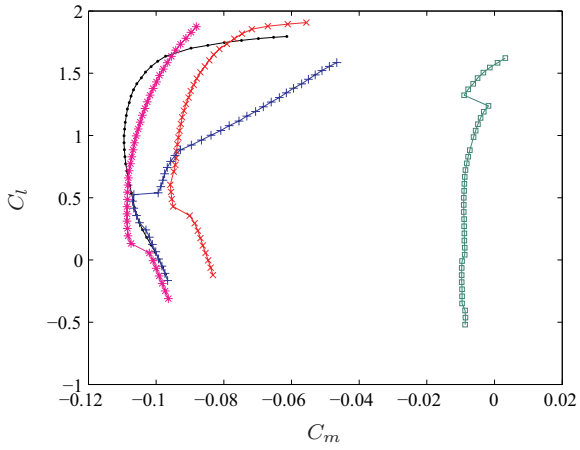
Figure 7.5: Geometrical profiles of identified designs of interest

Figure 7.5 features the geometrical profiles of the documented solutions in comparison to the NLF0416. The profile geometry of the identified solutions echo the conclusions derived from Figure 7.4, with the exception now that certain discrepancies in the airfoil geometry can clearly be attributed to prioritizing either design objective. For example, the design prioritizing f_1 is observed to be more symmetrical than the preferred design, with a very high upper surface curvature. This promotes the favourable pressure gradient to reduce drag losses due to the turbulent boundary layer. The design prioritizing f_2 is also more symmetrical, with the exception that the characteristic *s-shape* now appears on both the upper and lower surfaces, which balances the loading on the airfoil. Alternatively the design prioritizing f_3 is very asymmetrical, with a significantly larger aft camber. The larger aft camber undoubtedly serves to increase the amount of lift generated.

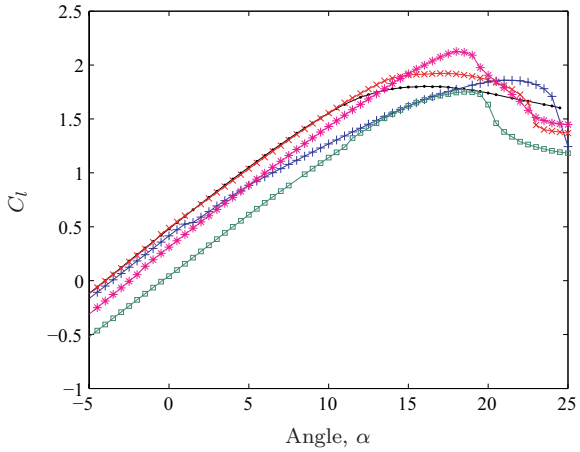
Figure 7.6 compares the designs of interest with the NLF0416 through aerodynamic performance curves. Figure 7.6(a) shows the drag polar of the identified airfoils, where the bucket region close to the cruise condition is clearly visible. For the min f_1 design, the drag bucket is more pronounced at the expense of larger drag values outside the design range. This is an undesirable feature of the airfoil, as it considerably reduces its domain of application. The pitching moment curve shown in Figure 7.6(b) clearly shows the favourable characteristics of the min f_2 design, yet its poor performance in drag and lift generation (see Figure 7.6(c)) clearly deters the selection of this airfoil for multi-mission flight. Alternatively, the preferred design and the min f_3 design offer reasonable performance in all areas which encourages their use for a multi-mission application.



(a) Drag polar



(b) Pitching moment curve



(c) Lift curve

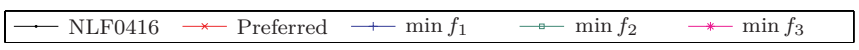


Figure 7.6: Performance comparison of identified designs of interest

Extended Application

To further refine the concept of a multi-mission design philosophy, an additional three objectives are implemented. This extension to the UAV airfoil design case-study was introduced in Wickramasinghe *et al.* [2010] for use with a many-objective PSO algorithm [see Wickramasinghe and Li, 2009]. The additional objectives cover a number of mission phases which are critical to the success of a multi-mission platform.

The design objective f_1 focused on minimizing the drag during flight which is essential to maximizing the range of the UAV. It is however also prudent that the airfoil obtain a high lift-to-drag ratio (or aerodynamic efficiency) for climbing, as well as increased flight endurance. Maximum endurance is a prerequisite for UAV which are expected to perform automated missions for an extensive period. For each candidate solution, the incidence angle is floated to determine the minimum drag-to-lift ratio, such that

$$f_4 = \min C_d/C_l^{3/2} \text{ at } Re = 4 \times 10^6,\ M = 0.3. \tag{7.4}$$

While the design objective f_3 focused on increasing the maximum lift of the airfoil, no reference is specified to maximizing the amount of generated lift in other areas of the flight envelope. The resulting deficiency is clearly evident from Figure 7.6(c), where the $\min f_3$ generates a greater maximum lift at the expense of poor lift generation at lower angles of incidence. During descent and approach conditions, a high lift value is beneficial as it constitutes towards an increased lift-induced drag, which is essential for landing. The angle of incidence is fixed at $\alpha = 5°$, which is regarded as a typical incidence angle during approach flight, such that

$$f_5 = 1/C_l^2 \text{ at } \alpha = 5°,\ Re = 2 \times 10^6,\ M = 0.15. \tag{7.5}$$

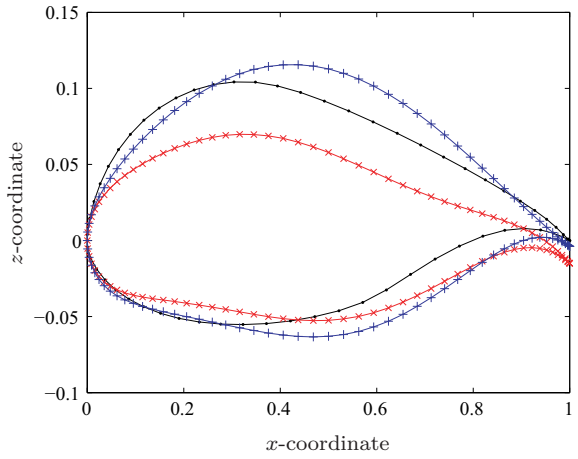
Providing optimal performance in the approach condition should not be at the expense of massive flow separation or leading-edge boundary layer transition (x_{tr}). In this case, it is sought to maintain a smooth flow-field during approach flight by maximizing the laminar portion of the upper surface of the airfoil, such that

$$f_6 = 1/x_{tr} \text{ at } \alpha = 5°,\ Re = 2 \times 10^6,\ M = 0.15. \tag{7.6}$$

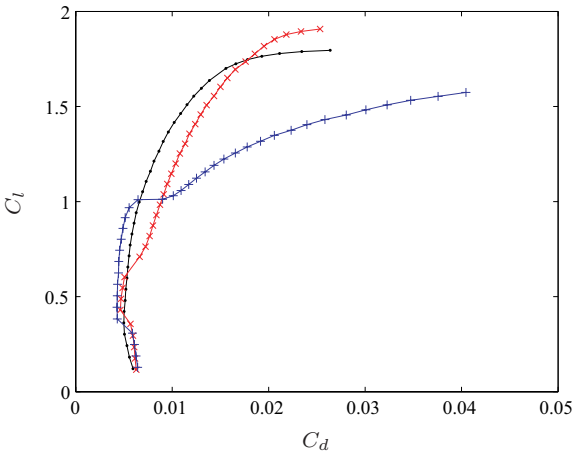
Results of the extended optimization case-study are presented in Wickramasinghe *et al.* [2010]. This study compares the user-preference framework with a conventional EMO framework and confirms the necessity of utilizing a reference point in a many-objective environment. The addition of three objectives ultimately changed the preferred compromise, resulting in a new identified spread of preferred designs. Table 7.3 consolidates the results of the extended optimization study.

Design	f_1	f_2	f_3	f_4	f_5	f_6
NLF0416	0.00516	0.00982	0.00308	0.00606	0.92314	0.65460
Preferred	0.00431	0.00968	0.00320	0.00581	0.89235	0.51930
min f_1	0.00407	0.01713	0.00287	0.00726	0.91924	0.64170
min f_2	0.00559	0.00413	0.00338	0.00671	0.96098	0.57380
min f_3	0.00412	0.03984	0.00239	0.00472	0.88731	0.70100
min f_4	0.00408	0.02883	0.00301	0.00404	0.97161	0.62610
min f_5	0.00604	0.01968	0.00276	0.00523	0.58638	0.64360
min f_6	0.00536	0.01145	0.00323	0.00477	0.93441	0.45250

Table 7.3: Results of the extended multi-mission airfoil optimization study



(a) Geometrical profile



(b) Aerodynamic drag polar

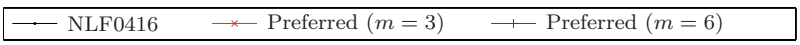


Figure 7.7: Comparison of the NLF0416 with the $m = 3$ and $m = 6$ preferred designs

From an initial observation of Table 7.3, it is evident that designs which exhibit improvement over all objectives have no longer been identified, owing largely to the increased complexity of the problem. Similar improvements (with reference to Table 7.2) have been identified by each min f_i design for the respective objective f_i. It is shown that the *revised* preferred design provides improvement for all objectives, at the expense of a reduction in the maximum lift coefficient (i.e. f_3). This suggests that the f_3 objective is driving the design and obtaining a high aerodynamic efficiency for nominal flight conditions hinders the manoeuvring performance of the airfoil.

Figure 7.7 features the original preferred design (for $m = 3$) and the revised preferred design (for $m = 6$). The revised profile is thicker and has a higher curvature over the upper surface, and the aft camber is less significant. Figure 7.7(b) shows that the performance region is no longer localized to a specific drag bucket as was evident for the original preferred design. This makes the selection of the revised preferred design more attractive for a multi-mission application, as a higher aerodynamic efficiency is observed across the flight envelope, notwithstanding the poorer performance at higher angles of incidence. It is observed that a multi-point optimization strategy is probably required to avoid a localized improvement of the drag at the expense of higher drag in off-design points.

7.2 Supersonic Nozzle Design

Flow conditions within supersonic nozzles are extremely sensitive to variations in profile geometry as well as ambient conditions and thus are an ideal candidate for shape optimization. Nozzles are designed to suit a broad range of engineering applications, such as rockets, turbines, supersonic wind tunnels, etc. Dependent on the application, the design philosophy employed will vary. For example, rockets require shorter nozzle lengths with rapid expansion to minimize weight, whereas supersonic nozzles require high quality uniform flow within the test section [Pasquale *et al.*, 2010]. The present work aims at understanding the trade-off between the reduction in length of the nozzle and the uniformity of the flow at the test section for a specified discharge Mach number.

Geometry Parameterization

An axisymmetric nozzle of circular cross-section is considered, such that the flow is symmetrical about the mid-plane. The geometry of the upper-half of the nozzle is shown in Figure 7.8. Air enters in from the reservoir inlet of area A_i. The flow within the converging section of the nozzle L_c is purely subsonic (i.e. $M < 1$) until the throat A_t where the flow is sonic (i.e. $M = 1$). Thereafter, the flow becomes supersonic (i.e. $M > 1$)

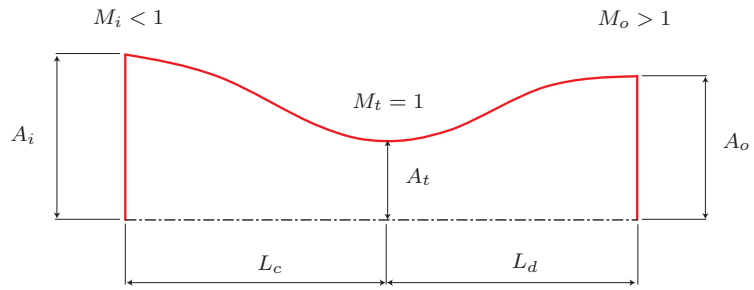


Figure 7.8: Upper-half of the nozzle geometry symmetrical with respect to the mid-plane

and accelerates through the diverging section L_d until it reaches the required discharge Mach number M_o at the outlet A_o. The nozzle length is fixed to unity and all geometrical quantities are made dimensionless with respect to the throat area A_t.

The uniformity of the supersonic flow at the outlet is evidently much more sensitive to the divergent section rather than the convergent section upstream of the throat section. The solid wall of the nozzle is therefore represented as a composition of two curves – whereby the convergent section is fixed and the divergent section is parameterized. Construction of the convergent section and throat geometry is important to ensure uniform flow at the throat, achieved through a smooth and gradual acceleration of the flow through the convergent section. The Mach-area relationship of one-dimensional isentropic flow theory is used to determine the inflow area:

$$\left(\frac{A}{A_t}\right)^2 = \frac{1}{M^2}\left[\frac{2}{\gamma+1}\left(1+\frac{\gamma-1}{2}M^2\right)\right]^{\frac{\gamma+1}{\gamma-1}} \tag{7.7}$$

where $\gamma = 1.4$ is ratio of specific heat for air. The area ratio between the inflow and the throat has been chosen such that the inlet Mach number is $M_i = 0.3$. As suggested by Macabe *et al.* [1967], the converging geometry is expressed via a composition of hyperbolic curves of the form:

$$y^2 = 1 + \frac{x^2}{\rho_c} \tag{7.8}$$

where ρ_c is the radius of curvature of the nozzle wall. Figure 7.9 features the convergent section geometry for a unit throat area. All the quantities required to define the converging portion of the nozzle have been chosen as a fixed multiple of the throat span, such that $A_i = 1.4266 \cdot A_t$ and $L_c = 3 \cdot A_t$, for an axisymmetric geometry.

The divergent section of the nozzle geometry is parameterized and constructed using a piecewise cubic Hermite interpolating polynomial. This is a shape preserving cubic spline function, defined by a series of control points (see Section 2.2.2). Hermite curves favourably do not exhibit oscillations unlike conventional spline methods. An additional benefit of

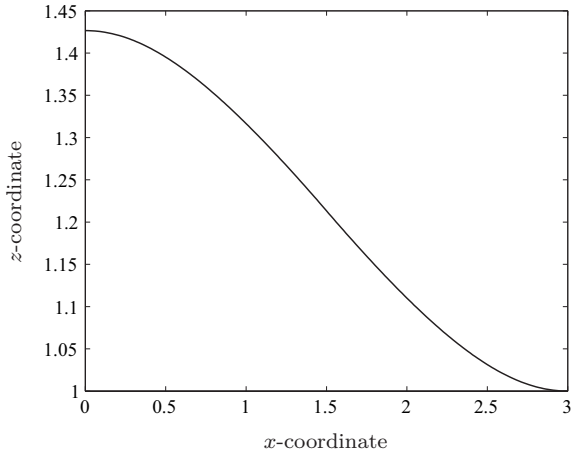


Figure 7.9: Normalized convergent section of the nozzle geometry

shape preserving curves is the ability to provide sufficient shape flexibility whilst easily enforcing the physical constraint of a continuously diverging section. Figure 7.10 illustrates the design of the divergent geometry, which is achieved by constructing a Hermite curve to represent the *expanding* and *levelling* sections of the supersonic nozzle.

Using isentropic flow theory as per Eq. (7.7), the outlet area is defined to ensure a discharge Mach number of $M_o = 2$, such that $A_o = 1.2989 \cdot A_t$. The Hermite curve is modelled by five control points which are horizontally and vertically translated to produce curves of varying degrees of curvature. The control points are illustrated in Table 7.4. The corresponding design variables are outlined in Table 7.5. The location of the throat L_c essentially measures the ratio between the converging and diverging lengths of the nozzle.

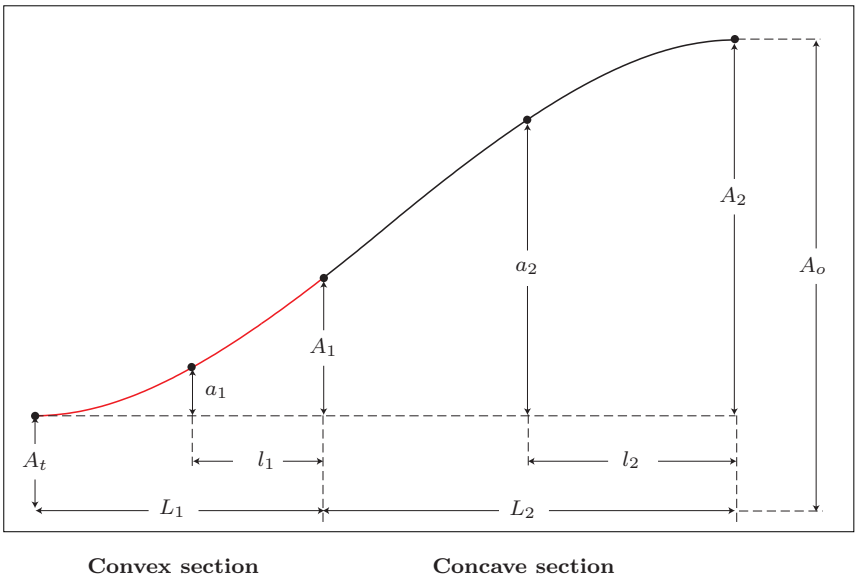


Figure 7.10: Nomenclature of the divergent section geometry

x-coordinate	z-coordinate
L_c	A_t
$L_1 - l_1$	$A_t + a_1$
L_1	$A_t + A_1$
$(L_1 + L_2) - l_2$	$A_t + a_2$
$L_1 + L_2$	$A_t + A_2$

Table 7.4: Hermite control points for supersonic divergent nozzle

This parameter is subsequently used to determine all important geometric quantities which are measured with respect to the throat area. The remaining variables define the control points, necessary to construct the divergent geometry.

Flow Solver

While the boundary layer and cross flow effects could represent a limiting issue for this application, in the present work, viscous effects are not considered and the flow is governed by the axisymmetric inviscid Euler equations. The general purpose finite volume code ANSYS Fluent is used. The axisymmetric inviscid Euler equations are solved using a pressure-based formulation with second-order spatial discretization, which is well suited for the computation of high Mach number flows. The operating pressure is set at zero, and the pressure and temperature ratios between the pressure inlet and outlet are specified through the gauge pressure. The two-dimensional computational grid is composed of a 200×50 body-fitted structured mesh, as shown in Figure 7.11. In the interest of robust and efficient convergence rates, Full Multi-Grid (FMG) initialization is employed, with coarsening of the grid to 10 cells. In the FMG initialization process, the Euler equations are solved using a first-order discretization to obtain a flowfield approximation before submitting to the full iterative calculation.

Variable	Description	Lower bound	Upper bound
L_c	Throat location	0.2	0.8
L_1	Convex section length	$0.1 \cdot (1 - L_c)$	$0.6 \cdot (1 - L_c)$
A_1	Convex section radius	$0.1 \cdot (A_2)$	$0.5 \cdot (A_2)$
a_1	Convex mid-length radius	$0.1 \cdot (A_1)$	$0.9 \cdot (A_1)$
a_2	Concave mid-length radius	$0.1 \cdot (A_2 - A_1)$	$0.9 \cdot (A_2 - A_1)$
l_1	Convex mid-length location	$L_1/2$	$L_1/2$
l_2	Concave mid-length location	$L_2/2$	$L_2/2$

Table 7.5: Design variable ranges for the supersonic nozzle case-study

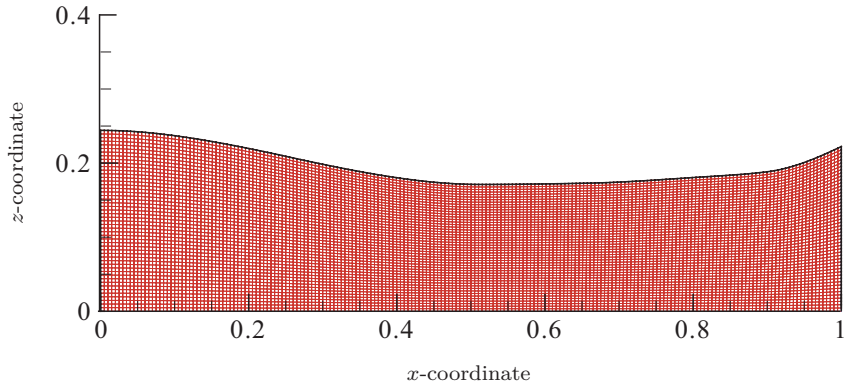


Figure 7.11: Example computational grid used during optimization process

Problem Formulation

The aim of the optimization case-study is to quantify the influence of nozzle length reduction on the uniformity of the Mach number at the outlet (i.e. the test section). The multi-objective problem is formulated by defining two conflicting objective functions:

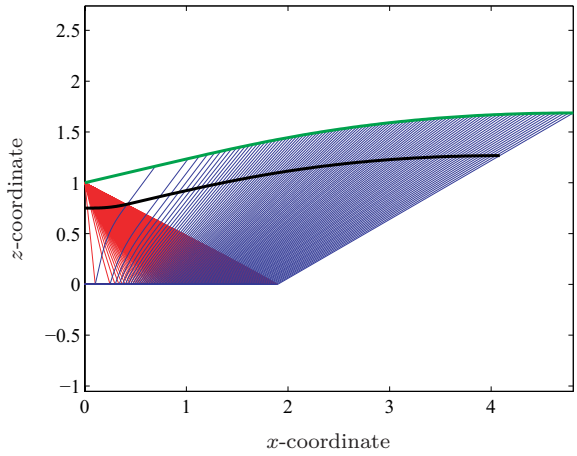
$$f_1 = \min \frac{1}{A_t}, \tag{7.9}$$

$$f_2 = \min \sqrt{\frac{1}{q} \sum_{j=1}^{q} (M_j - M_o)^2}. \tag{7.10}$$

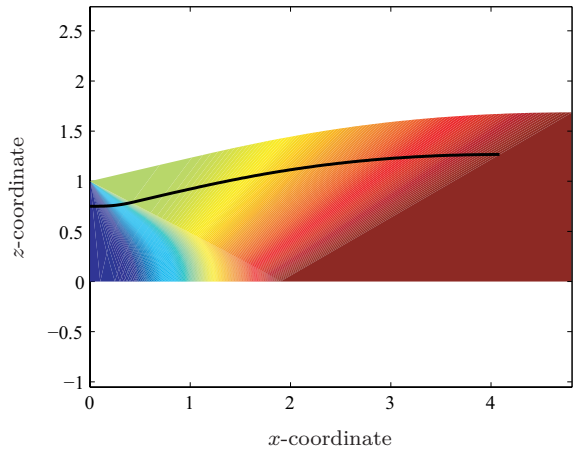
The objective f_1 is simply the inverse of the throat span, since it is related to the nozzle length. The second objective f_2 is the root mean square of the deviation of the local Mach number M_j of the j-th cell of the outlet boundary (discretized by q nodes), with respect to the design discharge Mach number.

The Reference Point Design

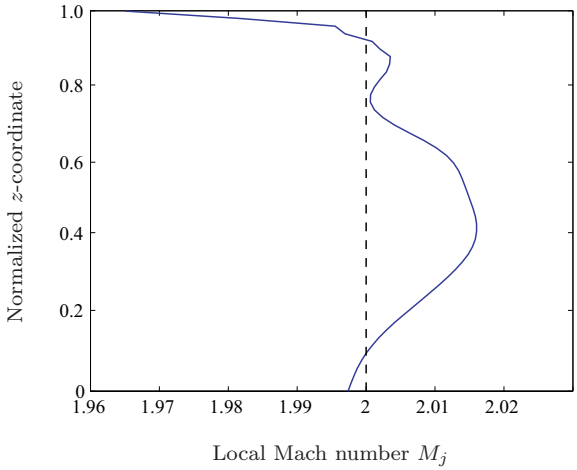
Prior to the advent of optimization techniques, nozzle design was achieved using the two-dimensional Method Of Characteristics (MOC) [Anderson, 2001]. MOC and its application to nozzle design is based on the linearization of an exact Prandtl-Meyer solution for supersonic corner flow. The method evaluates the interactions between a series of positive and negative characteristic Mach waves that propagate from two corner sources of a simplified nozzle shape. The limitation of MOC is that characteristic lines only exist in supersonic flow regimes and the method can therefore only be used for the design of the supersonic nozzle section downstream of the throat. This is not a serious problem for nozzle design, since the uniformity of the outlet flow is much more sensitive to the diverging section, rather than the converging section upstream of the throat. MOC is also limited in the sense that it cannot be extended to account for viscous effects and it provides no insight



(a) MOC design illustrating Mach wave intersects



(b) MOC design illustrating contours of Mach number



(c) Outlet flow as predicted by the Euler solver

Figure 7.12: Reference point geometry generated via the Method of Characteristics

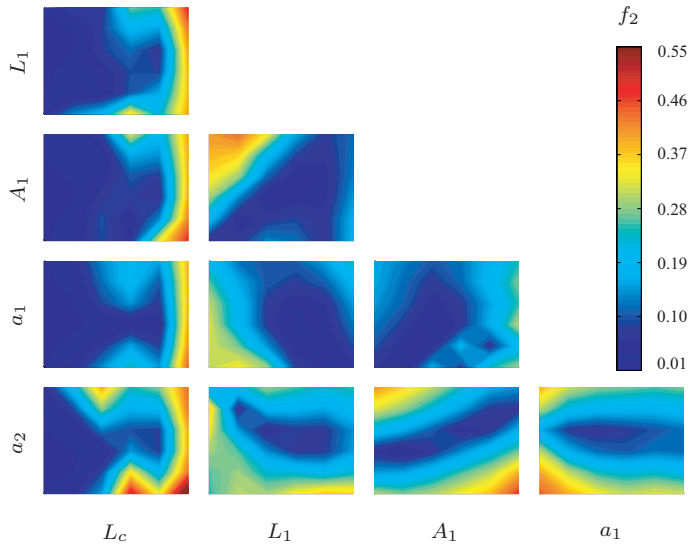


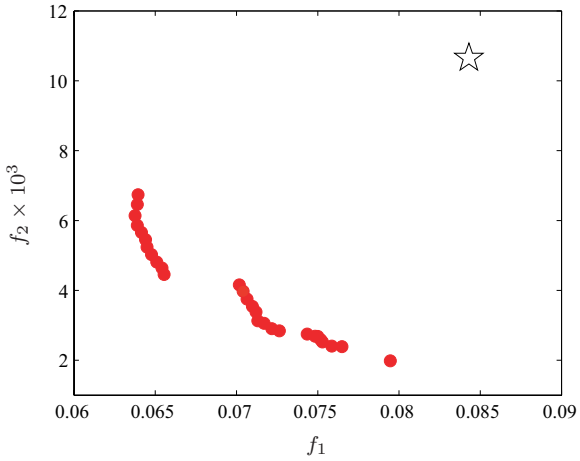
Figure 7.13: Two-dimensional contour projections of objective f_2

into the trade-off between flow quality and the nozzle length [Pasquale *et al.*, 2010]. Despite the deficiencies of MOC, it can provide a feasible reference point geometry which may be utilized by the optimization framework to identify designs of interest.

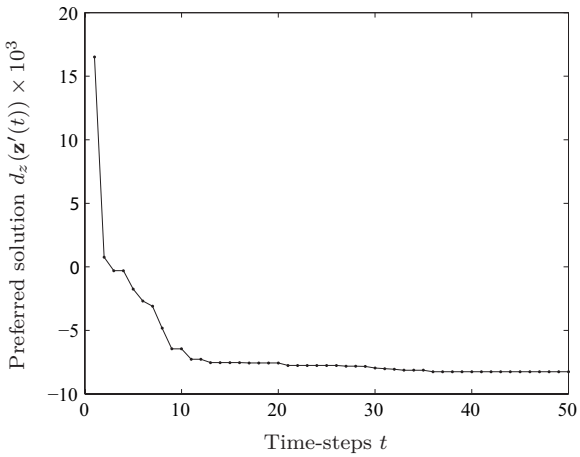
The analytical MOC (as shown in Figure 7.12) is used to generate the reference point divergent geometry based on a discharge Mach number of $M_o = 2$. The nozzle coordinates are expressed via the unit throat length and can subsequently be normalized via the nozzle length to be consistent with the problem formulation. The design geometry is characterized by the black line, as this refers to the shortest nozzle length required to achieve uniform flow at the outlet while maintaining zero curvature at the throat. The divergent geometry generated by MOC is subsequently meshed and solved using the Euler flow solver to generate Figure 7.12(c), used to calculate the reference value for f_2.

Variable Screening

The nozzle geometry is parameterized by a Hermite spline where the control points are representative of important geometrical features. This provides an ideal case-study for a variable screening study, to determine the elementary effect of each control point. To assess the design space topology the visualization technique introduced in Section 5.4.1 is first utilized. Figure 7.13 features the two-dimensional contour projections of the $n = 5$ dimensional design space. The multi-modality of the design space is fairly evident and it is clearly shown that each variable has non-linear interactions with all other variables. No meaningful conclusion could hence be derived from a screening study, other than ensuring that sufficient time is allocated to the swarm to navigate the landscape.



(a) Identified non-dominated solutions



(b) Progress of the most preferred solution

Figure 7.14: Optimization results of the supersonic nozzle design case-study

Optimization Results

The first objective is analytical and the Euler solver is also fairly economical, due to the fact that the mesh is not required to be dense and there is no added expense of computing the effects of viscosity. For this level of fidelity the construction of Kriging models is not deemed necessary. A swarm population of $N = 100$ particles is flown for $t_{\max} = 50$ time-steps. At each time-step, the swarm is partitioned across four parallel processors, yielding an average computational time per time-step of approximately twenty minutes (Intel Core2 Duo CPU). The reference point is obtained by computing the objective values for the MOC geometry using the flow solver (see Figure 7.12(c)) and a solution spread of $\delta = 5 \times 10^{-3}$ is specified.

Figure 7.14(a) features the final set of non-dominated solutions. The non-dominated front is discontinuous and three subsets of designs are identified. It is shown that the

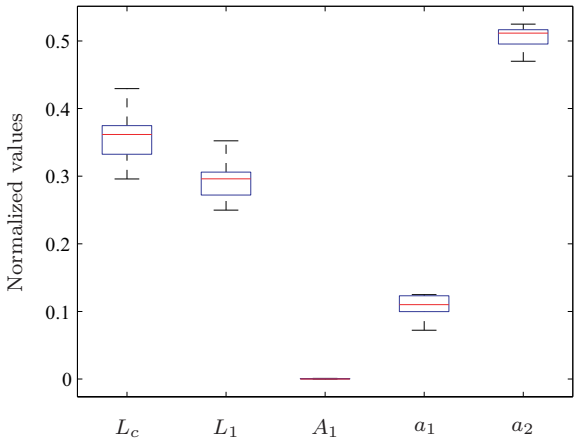


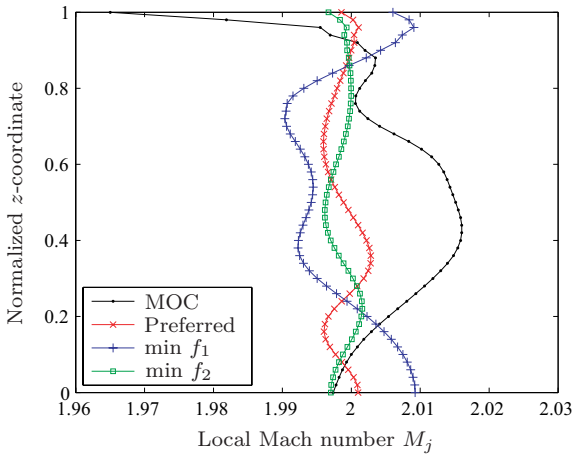
Figure 7.15: Normalized box-plots of variable ranges of the identified solutions

preferred compromise is biased towards objective f_2. This is expected since the MOC analysis estimates the most uniform design for the shortest nozzle length. Despite this the MOC geometry is still dominated and the UPMOPSO algorithm identified designs which improve on flow uniformity at a reduced nozzle length. The progress of the most preferred design is featured in Figure 7.14(b). As is a recurring observation, significant improvement occurs during exploration and convergence is clearly obtained within the allocated time.

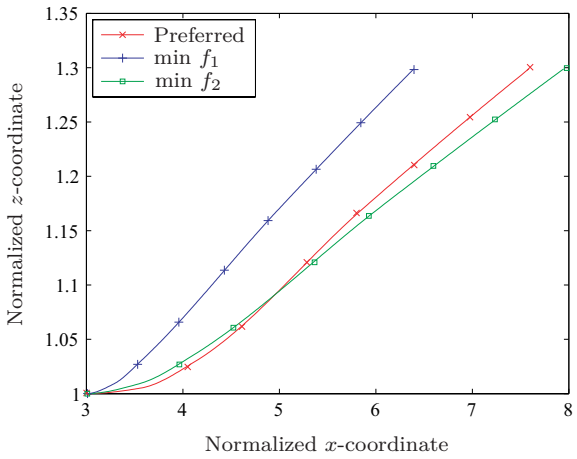
Final Designs

The identified non-dominated solutions are described using the box-plot shown in Figure 7.15. The objective f_1 is solely dependent on the throat location L_c and the box-plot illustrates that designs lie within 30 to 40% of the allocated range. This is not an intuitive result as L_c is inversely proportionate to f_1 and thus lower values of L_c result in larger nozzle lengths. This however signifies that shorter nozzle lengths will disrupt the uniformity of the flow at the outlet, deviating from the preferred compromise. Another interesting observation is that the convex radius A_1 and mid-length radius a_1 have very low magnitudes. The concave mid-length radius a_2 lies within 50% of the allocated range, which results in a quasi-linear concave section. This evidently restricts the formation of expansion waves thereby resulting in highly uniform flow at the outlet.

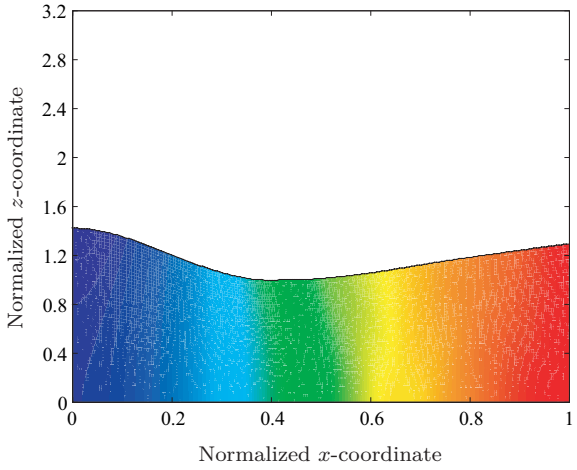
Figure 7.16(a) features the Mach number distributions of the preferred designs. The identified preferred designs offer a significantly more uniform flow at the exit over the MOC design. The corresponding nozzle shapes are shown in Figure 7.16(b). As concluded from the box-plot, the divergent section is quasi-linear, with a relatively short convex section and a linear concave section. The preferred geometry seems to deviate slightly from the other two designs which verifies the discontinuity observed in the non-dominated front.



(a) Outlet Mach number distribution



(b) Divergent nozzle shape



(c) Contours of Mach number for preferred design

Figure 7.16: Geometrical and performance comparison of identified nozzle designs

7.3 Transonic Airfoil Shape Optimization

This section describes a design framework to mitigate the adverse effects of shock formation and boundary layer separation through airfoil shape optimization. The work in this sections has been previously published in Carrese *et al.* [2011c]. Shape optimization of transonic airfoils is employed to limit shock drag losses and reduce shock-induced boundary layer instability at the design Mach number and lift coefficient. This often occurs at the expense of excessive pitching moments due to aft loading and performance degradation under off-design conditions. To overcome this inadequacy, robust optimization is generally performed for a wide operational spectrum.

Transonic design problems were traditionally solved using hodograph methods [see Drela and Giles, 1987] and more recently, the weighted-sum methodology (refer to Section 3.5.1). The latter allows for a larger number of design conditions, however the weight terms are generally not known in advance, which can lead to the optimizer exploiting undesirable regions of the design space. Utilizing the preference-based algorithm, focus is given exclusively to the region of the design space which compliments the performance compromise of an existing airfoil. This ensures that less strenuous effort is required in formulating the objectives, since the optimizer will attempt to improve the performance and mimic the design characteristics of an existing transonic airfoil.

Transonic Flow Solver

Potential flow and Euler methods are particularly suited to transonic design studies due to their quick and simple prediction of the wave drag (refer to Section 5.3.5). However the boundary layer also has a significant effect within the transonic regime, interacting with the shock wave to affect performance. In order to gain an appreciation of transonic flow it is necessary to also investigate the influence of viscosity.

In this study the finite volume RANS code of ANSYS Fluent is adopted (see Section 2.3.6). A coupled pressure-based numerical procedure is adopted with third-order spatial discretization to capture the occurring flow phenomena. The pressure-based coupled solver is a good alternative to density-based solvers when dealing with applications involving high-speed aerodynamics with shocks. The momentum equations and pressure-based continuity equation are solved concurrently, with the Courant-Friedrichs-Lewy number set at 200. The one-equation Spalart-Allmaras turbulence model [Spalart and Allmaras, 1992] is selected and turbulent flow is modeled over the entire airfoil surface. The C-type grid (as represented in Figure 7.17) stretches 25 chord lengths aft and normal of the airfoil section. Resolution of the C-grid is 460×65 providing an affordable mesh size of

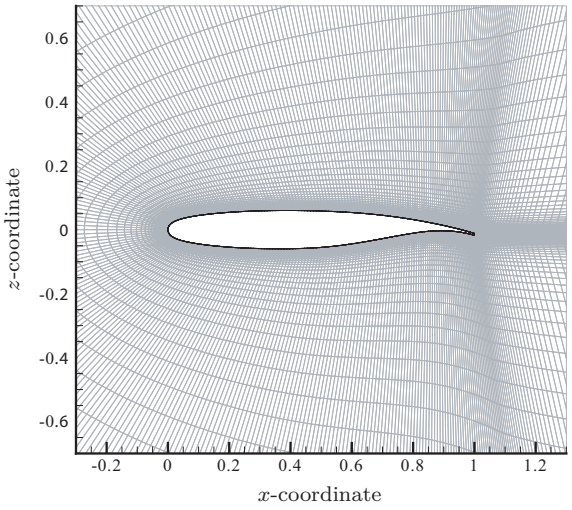


Figure 7.17: Airfoil c-type grid for transonic simulation

approximately 30,000 elements. The first grid-point is located 2.5×10^{-4} units normal to the airfoil surface resulting in an average y-plus value of 120. Standard wall functions are enabled since the viscous sub-layer is not resolved. In the interest of robust and efficient convergence rates, FMG initialization is employed, with coarsening of the grid to 30 cells.

Shape Parameterization and Qualitative Screening

Airfoils are parameterized using the popular PARSEC method for transonic airfoil design (see Section 2.2.3). The modification by Jahangirian and Shahrokhi [2009] is further implemented, to provide additional control over the trailing edge curvature. The KUPMOPSO algorithm has performed favorably for previous problems, since the smoothly correlated objective landscape compliments the Kriging interpolation method. Obtaining adequate prediction accuracy for airfoil performance coefficients in transonic viscous flow is generally more cumbersome. It is beneficial to construct the design space such that better performing regions are isolated, eliminating the probability of flow solver divergence.

SOMine is used to illustrate the PARSEC parameter sensitivity on the aerodynamic coefficients in a typical transonic environment. The PARSEC parameters may be restricted to conform to a specific family of airfoils to aid in the formation of the design space. This is achieved through inverse mapping of benchmark profiles that have been developed to perform favorably in transonic flow conditions. Four baseline airfoils are selected* and the desired variable(s) of interest are systematically varied to generate a qualitative description of their influence on the aerodynamic coefficients. Figure 7.18 features the selected baseline airfoils with the respective PARSEC parameter values shown in Table 7.6.

*The NASA SC (or supercritical) airfoils have been modified to incorporate a sharp trailing edge

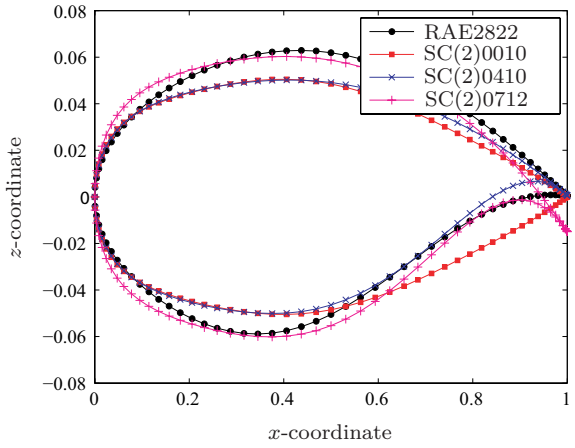


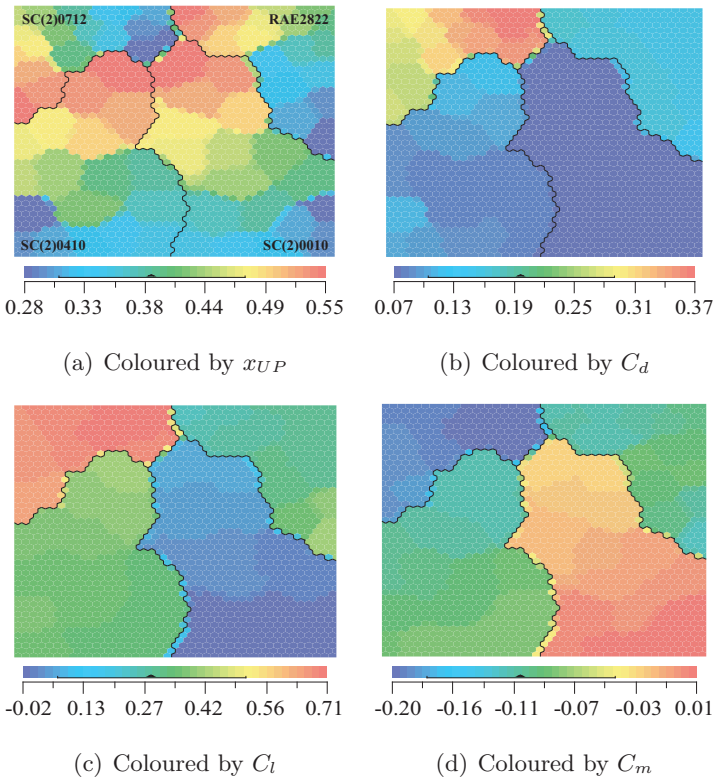
Figure 7.18: Baseline airfoils used in the sensitivity study

Airfoil	r_{LE}	α_{TE}	β_{TE}	x_{UP}	z_{UP}	$z_{xx_{UP}}$	x_{LO}	z_{LO}	$z_{xx_{LO}}$	$\delta_{\alpha_{TE}}$
RAE2822	0.0083	-0.1082	0.164	0.4324	0.0629	-0.4274	0.3436	-0.0589	0.7131	-0.0064
SC(2)0010	0.0119	0	0.2998	0.3974	0.0505	-0.3661	0.3974	-0.0505	0.3661	0
SC(2)0410	0.011	-0.1828	-0.017	0.4119	0.0502	-0.2637	0.376	-0.0501	0.4036	-0.0081
SC(2)0712	0.0164	-0.256	-0.0024	0.4081	0.0604	-0.3002	0.3695	-0.0601	0.514	0.1966

Table 7.6: Mapped PARSEC parameters of the baseline airfoils

For the following analysis, the Mach number is fixed at $M = 0.79$, and the angle of incidence at $\alpha = 0°$. The effect of varying the upper thickness location, x_{UP}, is shown in Figure 7.19. Every other variable is held constant across each of the four baseline configurations as shown in Table 7.6. Variable perturbations and the corresponding output coefficients are recorded and clustered according to the baseline airfoil.

The elementary effect of x_{UP} is easily identifiable through the SOM charts. Large variations to the drag coefficient are observed for medium to high-cambered airfoils, such as the NASA-SC(2)0712. In this case, lower-bound values of x_{UP} result in higher values of the drag coefficient, since this will effectively increase the upper surface curvature thereby resulting in a stronger shock-wave closer to the leading edge of the airfoil. Negligible variations to the drag coefficient are observed for the lower-cambered airfoils, which implies x_{UP} is a non-sensitive variable. On the contrary, lower-cambered airfoils are observed to be more sensitive to variations of x_{UP} when considering the lift and pitching moment characteristics. For the symmetrical SC(2)0010, an increment in x_{UP} is complemented with an increase in the lift coefficient due to the effective camber created. To guarantee the feasibility of incorporating the additional trailing edge design variable, the effect of varying $\delta_{\alpha_{TE}}$ is shown in Figure 7.20. Similar to the previous study, every other variable is held at the baseline values. The variable perturbations and the corresponding output coefficients are recorded and clustered according to the baseline airfoil.



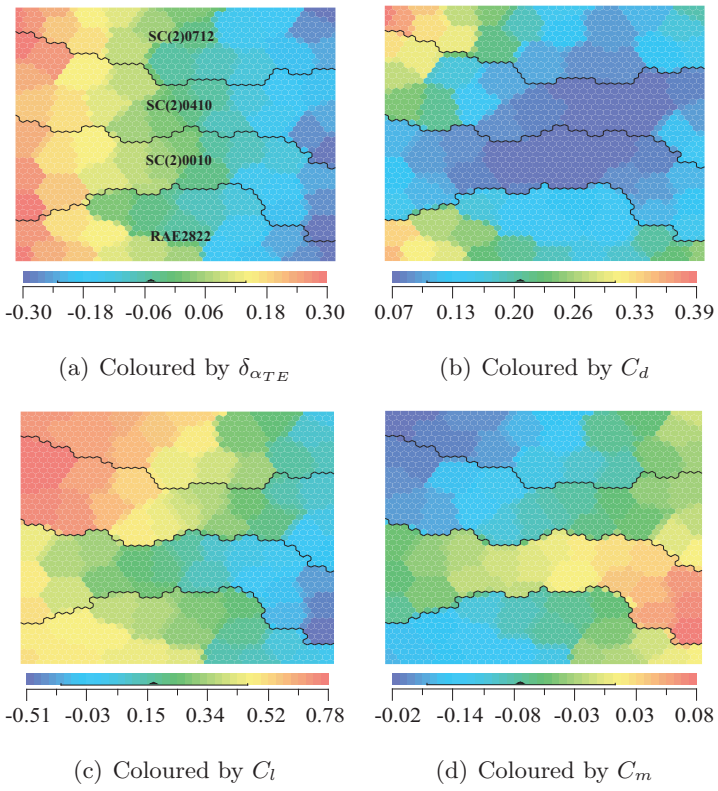
(a) Coloured by x_{UP} (b) Coloured by C_d

(c) Coloured by C_l (d) Coloured by C_m

Figure 7.19: Effect of varying x_{UP} on aerodynamic coefficients

In comparison to x_{UP} the SOM charts for variations in $\delta_{\alpha_{TE}}$ demonstrate its high sensitivity to all the output aerodynamic coefficients. Furthermore, for each baseline airfoil a diverse trend is observed. Focusing on the symmetrical SC(2)0010 airfoil, it is observed that as the boundaries of $\delta_{\alpha_{TE}}$ are approached a subsequent increase in the drag coefficient is obtained, owing to the effective increase in aft camber. This effective increase in aft camber also results in a lift increment of $\Delta C_l \approx \pm 0.50$. For the highly-cambered SC(2)0712 airfoil, as the lower bound value of $\delta_{\alpha_{TE}}$ is approached a significant reduction in the drag coefficient and pitching moment magnitude is observed with only a mild reduction in the lift coefficient. This is due to the reduction in aft loading due to the change in trailing edge curvature, as well as the reduction in upper surface curvature, which helps to reduce the strength of the shock-wave.

It is clear that variations in the PARSEC parameters and their effect on the airfoil aerodynamics are also due to other geometrical variables, e.g. thickness, camber, etc. Therefore, variations in trends are not a result of the elementary effect of a single variable, but rather the interactive effects with other variables. SOM charts can also be used to gather qualitative data on the interactive effect of a variable group (in this case limited to couple-interactions). Figure 7.21 features the interactive effect of the leading edge radius



(a) Coloured by $\delta_{\alpha_{TE}}$ (b) Coloured by C_d

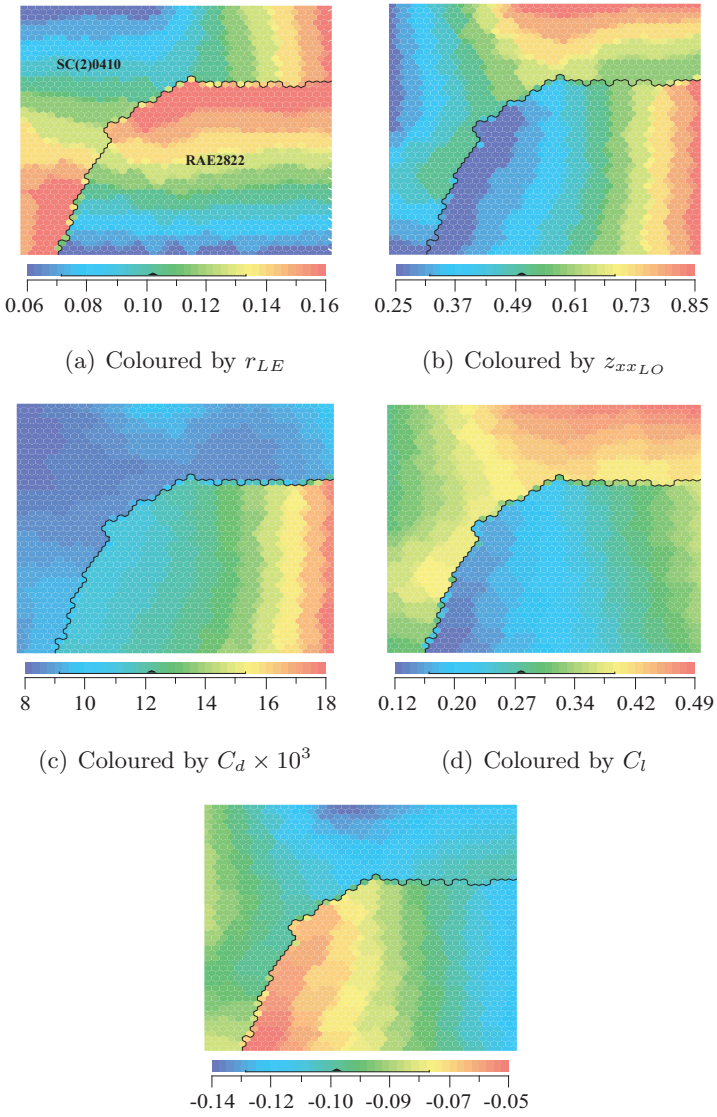
(c) Coloured by C_l (d) Coloured by C_m

Figure 7.20: Effect of varying $\delta_{\alpha_{TE}}$ on aerodynamic coefficients

r_{LE} and the lower surface curvature $z_{xx_{LO}}$ on the aerodynamic coefficients. Only two baseline airfoils are included to facilitate comprehension of the charts.

It is observed that for the RAE2822 airfoil, the aerodynamic properties in this case are heavily influenced by variations in the lower surface curvature $z_{xx_{LO}}$, since this effectively controls the degree of aft loading experienced on the airfoil, fundamental to transonic aerodynamics. The variations in the leading edge radius r_{LE} have a minor effect on the aerodynamic properties when interacting with $z_{xx_{LO}}$. Similar conclusions are drawn for the SC(2)0410 baseline airfoil, although it is now interesting to note that the interactive effect r_{LE} on the aerodynamic properties is no longer negligible. An increase in r_{LE} despite variation in $z_{xx_{LO}}$ will result in an increment in the drag, as a result of the pressure peak being shifted closer to the leading edge. The lift and moment coefficients are still predominantly dictated by $z_{xx_{LO}}$.

Such design variable sensitivity studies are useful in obtaining a thorough representation of the design landscape. Performing screening studies are advantageous to propose feasible design space boundaries which bypass poorly performing areas of the design space. Design space visualization is also beneficial to monitor the multi-modality of the variable-to-objective landscape in order to impose an appropriate computational budget. Table 7.7



(a) Coloured by r_{LE}

(b) Coloured by $z_{xx_{LO}}$

(c) Coloured by $C_d \times 10^3$

(d) Coloured by C_l

(e) Coloured by C_m

Figure 7.21: Interactive effect of varying r_{LE} and $z_{xx_{LO}}$ on aerodynamic coefficients

features the upper and lower boundaries for the subsequent optimization case-study. These boundaries have been selected based on a thorough screening study involving a statistical sample of a number of benchmark airfoils.

Problem Formulation and Reference Point

The parameterization method and transonic flow solver described in the preceding section are now integrated within the KUPMOPSO algorithm for an efficient airfoil design framework. The framework is applied to the re-design of the NASA-SC(2)0410 airfoil for robust aerodynamic performance. A three-objective constrained optimization problem is formulated, with $f_1 = C_d$ and $f_2 = -C_m$ for $M = 0.79$, $C_l = 0.4$, and $f_3 = \partial C_d/\partial M$ for the

Variable	Lower bound	Upper bound
r_{LE}	0.0063	0.0151
α_{TE}	0.2405(-)	0.0026(-)
β_{TE}	0.0655	0.2618
x_{UP}	0.3170	0.5250
z_{UP}	0.0497	0.0683
$z_{xx_{UP}}$	0.5135(-)	0.2393(-)
x_{LO}	0.2835	0.3418
z_{LO}	0.0603(-)	0.0478(-)
$z_{xx_{LO}}$	0.2535	0.8405
$\delta_{\alpha_{TE}}$	0.0080(-)	0.3696

Table 7.7: PARSEC parameter ranges for transonic optimization case-study

Airfoil	M	f_1	f_2	f_3
NASA-SC(2)0410	0.79	0.008708	0.1024	0.189625

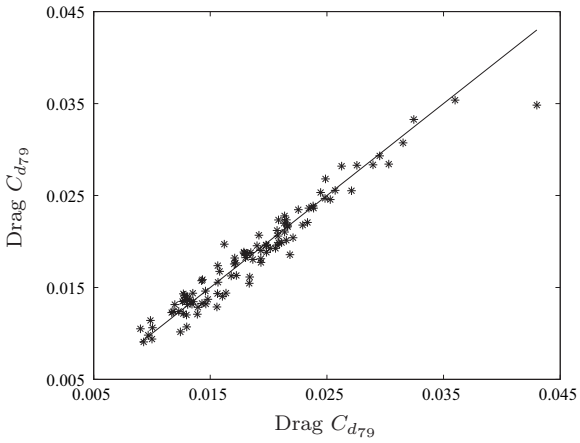
Table 7.8: NASA-SC(2)0410 airfoil results for the formulated objectives

design range $M = [0.79, 0.82]$, $C_l = 0.4$. The lift constraint is satisfied internally within the solver, by allowing Fluent to determine the angle of incidence required. A constraint is imposed on the allowable thickness, which is defined through the parameter ranges (see Table 7.8) as approximately 9.75% of chord. The reference point is logically selected as the NASA-SC(2)0410, in an attempt to improve on the performance characteristics of the airfoil, whilst still maintaining a similar level of compromise between the design objectives. The extent of the preferred region is controlled by $\delta = 5 \times 10^{-3}$.

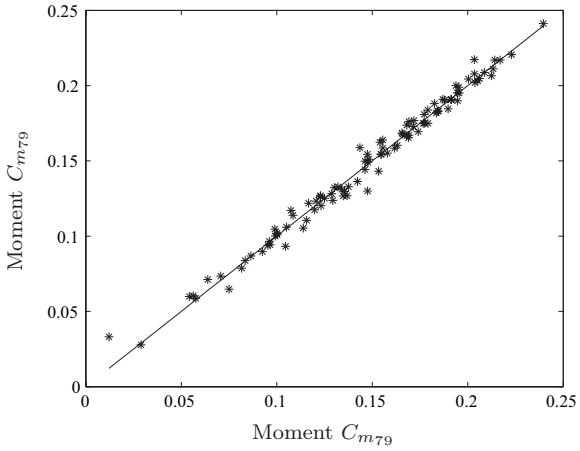
Model Construction and Quantitative Screening

Global Kriging models are constructed for the aerodynamic coefficients from a stratified sample of $N_K = 100$ design points[1]. This sampling plan size is considered sufficient in order to obtain sufficient confidence in the results of the subsequent design variable screening analysis. Whilst a larger sampling plan is essential to obtain fairly accurate correlation, the interest here is to quantify the elementary effect of each variable to the objective landscapes. The global Kriging models are initially trained via cross-validation. Illustrated in Figure 7.22 are the CV curves for the Kriging models. The subscripts to the aerodynamic coefficients refer to the respective operational Mach numbers.

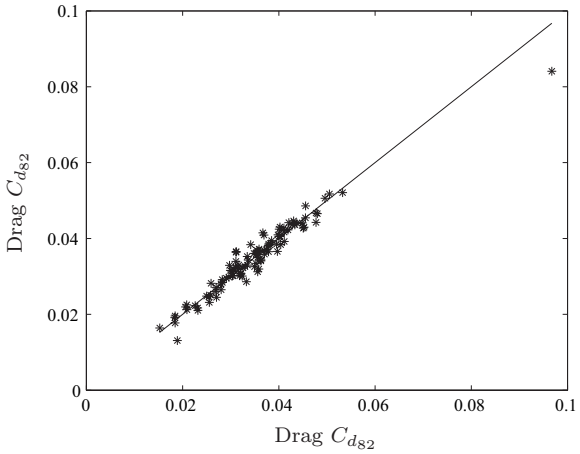
[1]This follows the suggestion of Forrester *et al.* [2008], that the computational expense of building a Kriging dataset should be approximately one-third of the total computational budget



(a) Training sample for C_d at $M = 0.79$



(b) Training sample for C_m at $M = 0.79$



(c) Training sample for C_d at $M = 0.82$

Figure 7.22: CV plots for the constructed Kriging models

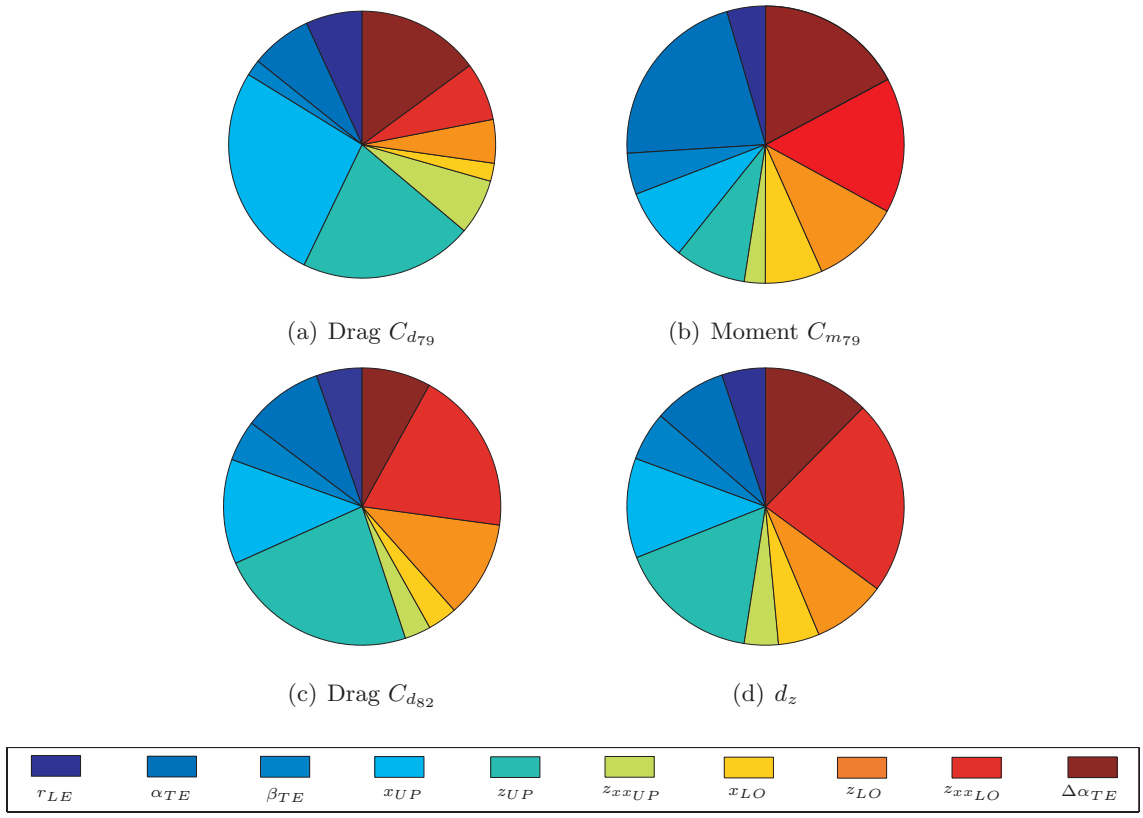
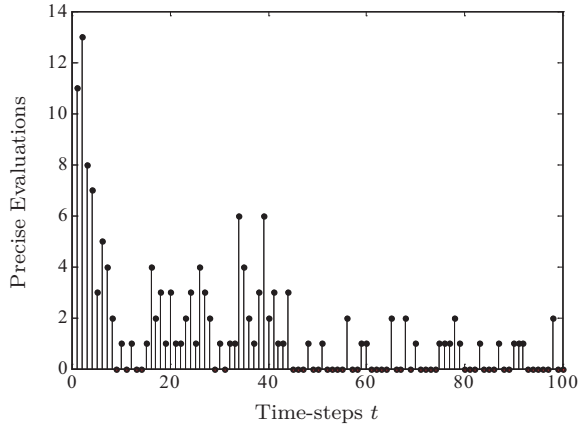


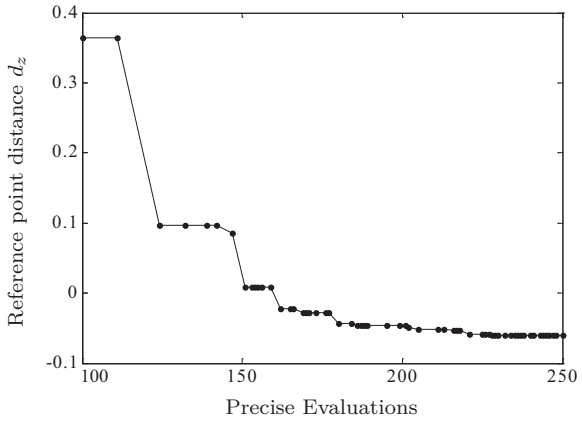
Figure 7.23: Quantitative illustration of variable influence on the aerodynamic coefficients

Figure 7.22 shows the constructed Kriging models reproduce the training samples with sufficient confidence, recording error margins of 2% to 4%. It is hence concluded that the Kriging method is very adept at modelling complex landscapes with only limited precise observations. In the previous section the variable sensitivity on a transonic landscape was qualitatively illustrated. While this analysis proved beneficial to construct the design space, the immediate effect of the variables on the objective landscapes and the preferred compromise was not assessed. To investigate these effects, a quantitative visualization analysis via the Morris method is conducted.

Figure 7.23 graphically shows the results obtained from the design variable screening study. It is immediately observed that the upper thickness coordinates has a relatively large influence on the drag coefficient for both design conditions. At higher Mach numbers the effect of the lower surface curvature $z_{xx_{LO}}$ is also significant. It is demonstrated however that the variables $z_{xx_{LO}}$ and α_{TE} have the largest effect on the moment coefficient - variables which directly influence the aft camber on the airfoil. These variables will no doubt shift the loading on the airfoil forward and aft resulting in highly fluctuating moment values. Similar deductions can be made by examining the variable influence on d_z shown in Figure 7.23(d). The variable influence on d_z is case-specific and entirely dependent on



(a) History of precise updates



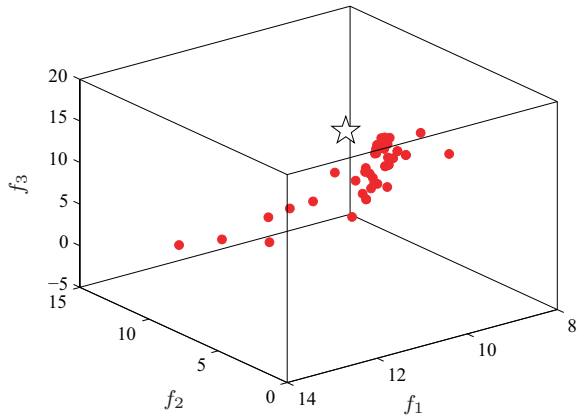
(b) Progress of the most preferred solution

Figure 7.24: KUPMOPSO performance for transonic airfoil shape optimization

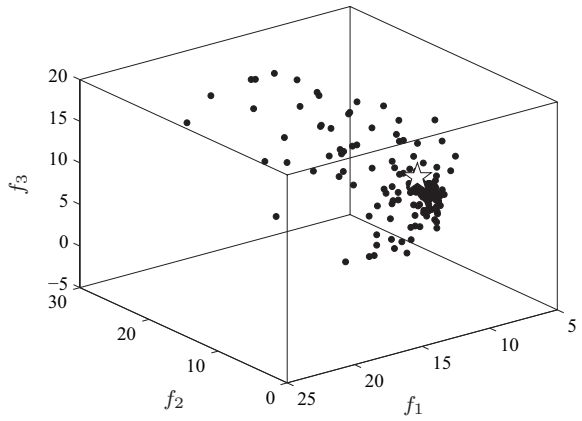
the reference point chosen for the proposed optimization study. Since the value of d_z is a means of ranking the success of a multiobjective solution as one single scalar, variables may be ranked by influence, which is otherwise not possible when considering a multiobjective array. Preliminary conclusions to the priority weighting of the objectives to the reference point compromise can also be made. It is observed that the variable influence on d_z is most closely resembling the the plots of the drag coefficients $C_{d_{79}}$ and $C_{d_{82}}$, suggesting that the moment coefficient is of least priority for the preferred compromise. It is curious to see that $\Delta\alpha_{TE}$ is important for all design coefficients, which confirms the results of the qualitative analysis and validates its inclusion in the optimization study.

Optimization results

A swarm population of $N = 100$ particles is flown to solve the optimization problem until the computational budget of 250 evaluations is breached. The objective space is normalized



(a) Scatter plot of all precise evaluations ($\mathbf{f} \times 10^3$)

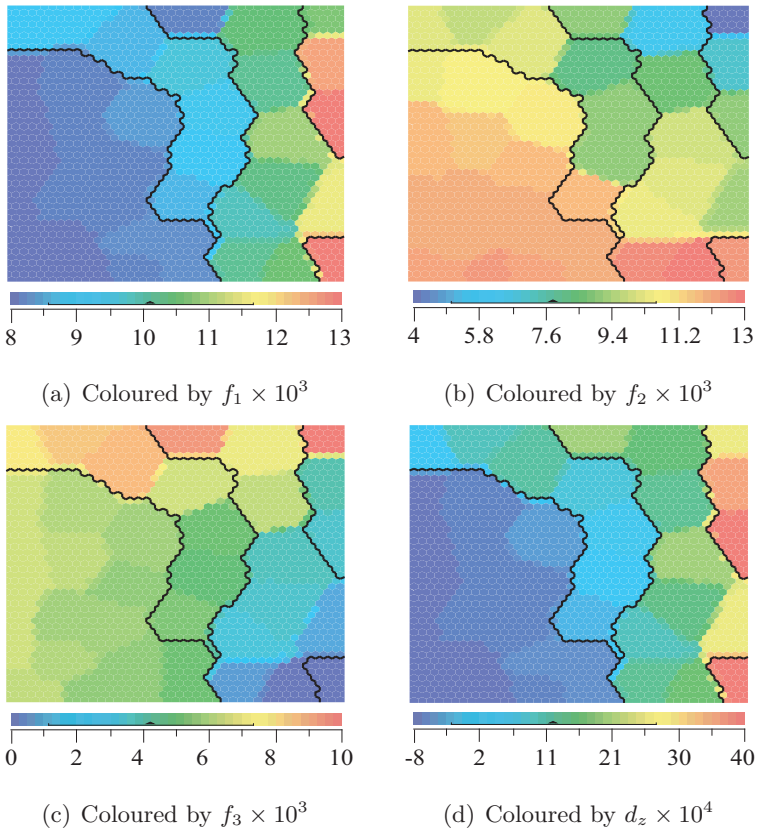


(b) Scatter plot of all precise evaluations ($\mathbf{f} \times 10^3$)

Figure 7.25: Precise evaluations performed and the resulting non-dominated solutions

for the computation of the reference point distance by $(f_{\max} - f_{\min})$. The sample of $N_K = 100$ design points was used to construct the initial global Kriging approximations for each objective. A further 150 precise updates were performed over $t \approx 100$ time-steps. As shown in Figure 7.24(a), the largest number of update points is recorded during the initial explorative phase. As the preferred region becomes populated and $\hat{s} \to 0$, the algorithm triggers exploitation and the number of update points steadily reduce.

The KUPMOPSO algorithm proves to be very capable for this specific problem. Figure 7.24(b) features the progress of the highest ranked solution (i.e. $\mathbf{z}'$) as the number of precise evaluations increase. The reference point criterion is shown to be proficient in filtering out poorer solutions during exploration, since only 50 update evaluations are required to reach within 15% of the final converged design and a further 50 evaluations to reach within 3%. Furthermore, no needless evaluations as a result of the lower-bound prediction are performed during the exploitation phase. This conclusion is further comple-



(a) Coloured by $f_1 \times 10^3$ (b) Coloured by $f_2 \times 10^3$

(c) Coloured by $f_3 \times 10^3$ (d) Coloured by $d_z \times 10^4$

Figure 7.26: Trade-off visualization via SOM charts for the transonic design problem

mented by Figure 7.25(a), as a distinct attraction to the preferred region is clearly visible. A total of 30 Pareto-optimal solutions were identified in the preferred region, which are shown in Figure 7.25(b).

Trade-off visualization

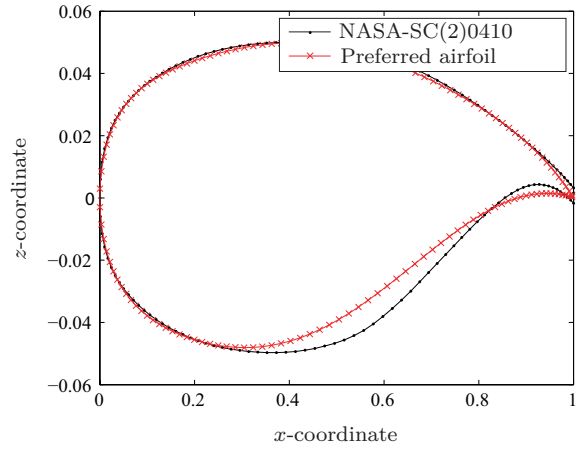
Figure 7.26 features the trade-off visualization analysis conducted using SOMine, where SOM charts are organized by six clusters according to ascending values of d_z. Following the SOM charts, it is possible to visualize the preferred compromise between the design objectives which are obtained. The chart of d_z closely follows the f_1 chart, which suggests that this objective has the highest priority.

Final designs

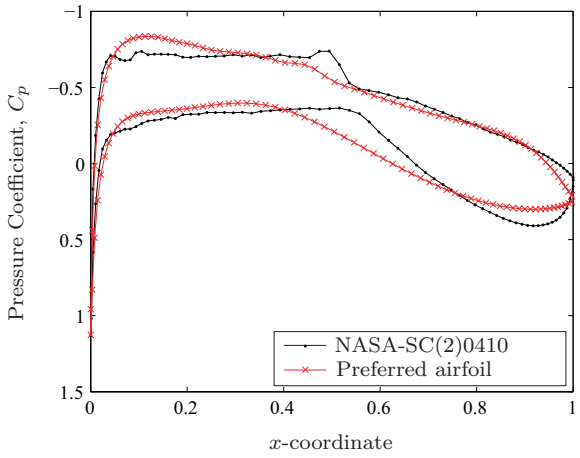
Table 7.9 shows the objective comparisons with the NASA-SC(2)0410. Interesting to note is that the most active objective is f_1, since the solution which provides the minimum d_z values also provides the minimum f_1 value. This implies that the reference point was situated near the non-dominated front boundary of f_1. Of the identified set of non-

Airfoil	f_1	f_2	f_3
NASA-SC(2)0410	0.008708	0.1024	0.189625
Preferred Design	0.008106	0.0933	0.168809
% Improvement	6.9	8.8	10.9

Table 7.9: Objective values for the preferred and reference designs with respective improvement



(a) Preferred airfoil geometry



(b) C_p distributions for $M = 0.79$, $C_l = 0.4$

Figure 7.27: Most preferred solution observed by the KUPMOPSO algorithm

dominated solutions, the largest improvements obtained in objectives f_2 and f_3 are 36.4% and 91.6% respectively over the reference point. The preferred airfoil geometry is shown in Figure 7.27(a). The preferred airfoil has a thickness of 9.76% chord and maintains a moderate curvature over the upper surface. A relatively small aft curvature is used to generate the required lift, at the same time reducing the magnitude of pitching moment. A performance comparison between the NASA-SC(2)0410 and the preferred airfoil at

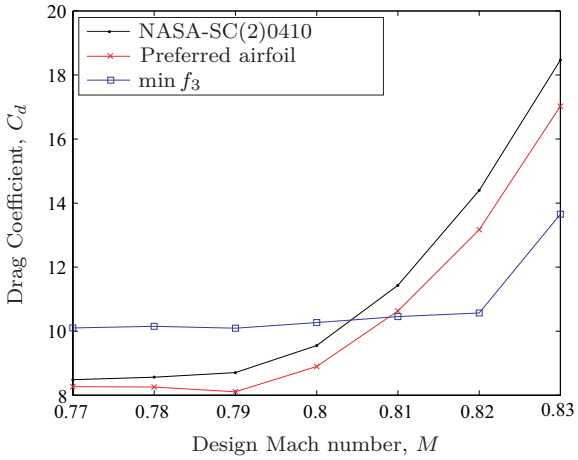


Figure 7.28: Drag-rise curves for $C_l = 0.4$

the design condition of $M = 0.79$ can be made from the surface pressure distribution of Figure 7.27(b). The reduction in C_d is attributed to the significantly weaker shock which appears slightly upstream of the supercritical shock position. The reduction in the pitching moment is clearly visible from the reduced aft loading.

Along with the improvement at the required design condition, the preferred airfoil exhibits a lower drag-rise by comparison, as shown in Figure 7.28. There is a notable reduction in the drag coefficient at the design condition of $M = 0.79$ (resulting in a negative drag-rise gradient), and the drag is recorded as lower than the NASA-SC(2)0410 even beyond the design range. Also shown is the design which provides the most robust performance (i.e. min f_3). The most robust design is clearly not obtained at the expense of poor performance at the design condition, which is an advantage of the preferred region.

7.4 Fuselage Cross-Sectional Design

Managing a discontinuous objective landscape poses a large problem for surrogate modelling. Since this algorithm focuses on modelling a preferred area of the design space, the influence of a discontinuous landscape is alleviated, since only a partial region is modelled. As the main illustration of this algorithm to manage discontinuous design problems, the multi-disciplinary design of a semi-monocoque fuselage enclosing a pressurized cabin and payload bay is considered. A fundamental design rule here is that any cross-sectional shape other than a circle is a stress compromise [Niu, 2002]. Any deviation from the circular shape forces the frames to carry a bending load (otherwise they are limited to maintaining the shape of the fuselage and breaking up the lengths of the longerons). However, competing drivers (primarily, the minimization of pressure drag by a reduction of cross-sectional

area and the maximization of passenger comfort by increasing certain cabin dimensions) routinely demand other shapes.

The classic alternative is the two-lobe cross-section. The upper and lower lobes are referred to as the *passenger* and *cargo* lobes respectively. The passenger cabin and the cargo bay are separated by the cabin floor, which carries tensile loads resulting from the pressurization, as well as the bending loads caused by the weight of the seats, passengers, etc. From a design optimization perspective, a parametric description capable of covering a broad range of two-lobe designs is therefore required.

Parameterization Model

For fuselage design, generating *families* of shapes which reproduce standard sections as well as multiple overlapping sections in a concisely parameterized manner is advantageous. This would enable the Pareto analysis of the trade-offs involved in deviations from circularity, mainly driven by the competing goals of structural weight, drag minimization and comfort. The study of the systematic description of such pseudo-circular fuselage cross-section shapes goes back to the earliest days of CAD [Polhamus *et al.*, 1959].

The right-hand half of the generic airliner fuselage cross-section in the $y-z$ plane is defined as the explicit function

$$y(z) = C_S \cdot \max\left[Y^{\mathrm{CAR}}(z), Y^{\mathrm{PAX}}(z)\right] + \Delta z,\ z \in [0,1], \tag{7.11}$$

where the coordinates $y(z)$ are a composite depiction of the individual shape descriptions, which use class functions from Kulfan [2008] (see Section 2.2.4) of the cargo lobe (Y^{CAR}) and the passenger lobe (Y^{PAX}).

$$Y^{\mathrm{CAR}}(z) = \begin{cases} C^{\mathrm{CAR}} z^{N_1^{\mathrm{CAR}}} \left(2R^{\mathrm{CAR}} - z\right)^{N_2^{\mathrm{CAR}}}, & z \in \left[0, 2R^{\mathrm{CAR}}\right] \\ 0 & \text{elsewhere.} \end{cases} \tag{7.12}$$

$$Y^{\mathrm{PAX}}(z) = \begin{cases} C^{\mathrm{PAX}} \left[z - \left(1 - 2R^{\mathrm{PAX}}\right)\right]^{N_1^{\mathrm{PAX}}} (1-z)^{N_2^{\mathrm{PAX}}}, & z \in \left[1 - 2R^{\mathrm{PAX}}, 1\right] \\ 0 & \text{elsewhere} \end{cases} \tag{7.13}$$

In its most flexible form, the cross-sectional shape is described by ten variables, each with a clear and intuitive meaning. The passenger lobe and the cargo lobe have variable radii: R^{PAX} and R^{CAR} respectively. The deviations from circularity are controlled by two exponents on each lobe: N_1^{PAX}, N_2^{PAX}, N_1^{CAR} and N_2^{CAR}. Additional flexibility is enabled by a scaling coefficient on each lobe (C^{PAX} and C^{CAR}) (see Figure 7.29 for an example). The section is normalized to a height of $z = 1$. The coefficient C_S and the offset Δz define the full size, correctly positioned section.

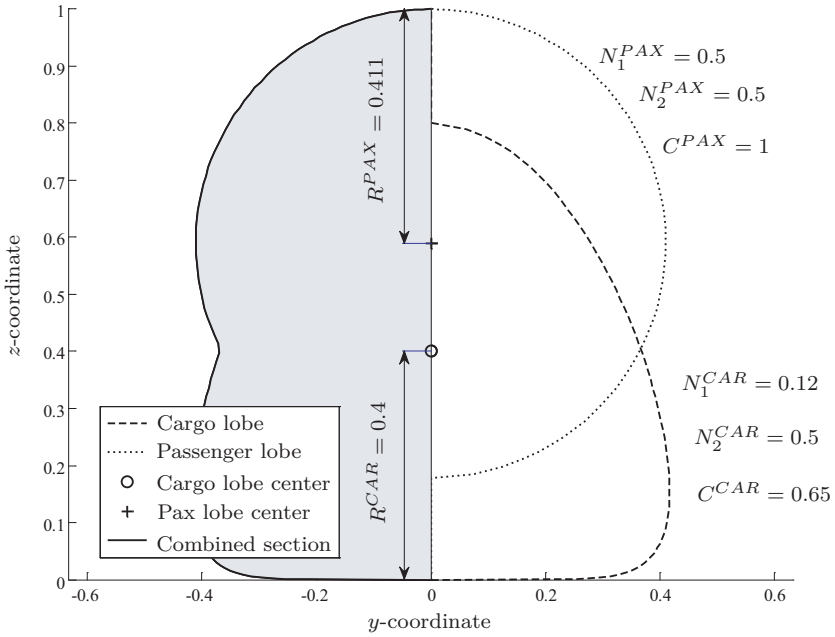


Figure 7.29: Approximation to the wing-to-body fairing area of the Embraer E145

Problem Formulation and Reference Point

The problem is defined as the design of a single-aisle fuselage cross-section which accommodates 95 percentile US male passengers seated six abreast, respecting the industry standard requirements in terms of aisle headroom, headroom under the overhead bins, overhead bin space, window seating, foot, shoulder and headroom. The cargo bay is designed to accommodate the standard LD3-45W container (see Figure 7.30). These conditional requirements are incorporated into a set of constraint points (marked by + symbols), which must all reside within the cross-section to ensure a feasible shape.

The design variable vector is formally defined as:

$$\mathbf{x} = \left\{ R^{\mathrm{CAR}}, N_1^{\mathrm{CAR}}, N_2^{\mathrm{CAR}}, C^{\mathrm{CAR}}, R^{\mathrm{PAX}}, N_1^{\mathrm{PAX}}, N_2^{\mathrm{PAX}}, C^{\mathrm{PAX}}, W_{\mathrm{S}}, W_{\mathrm{A}}, W_{\mathrm{AR}}, H_{\mathrm{B}} \right\} \quad (7.14)$$

where the definition and ranges of these variables are provided in Table 7.10. The number of constraint points that reside *outside* the cross-section is denoted as N_{out}, and hence the constraint $N_{\mathrm{out}}(\mathbf{X}, C_S, \Delta Z) = 0$ is formulated. The constraint is satisfied by the parametric model and not handled directly by the optimizer, yielding a first-order discontinuity. As KUPMOPSO is a gradient-free algorithm, this approach is permissible.

For a given input vector $\mathbf{x}$ it is sought to minimize the pressure drag created by the cross-section while still enveloping the constraint points. The objective f_1 is thus formulated as the cross-sectional (half-)area:

$$f_1(\mathbf{x}) = \min_{C_S, \Delta z} \int_0^1 y(z)dz \quad \text{subject to } N_{\mathrm{out}} = 0, \quad (7.15)$$

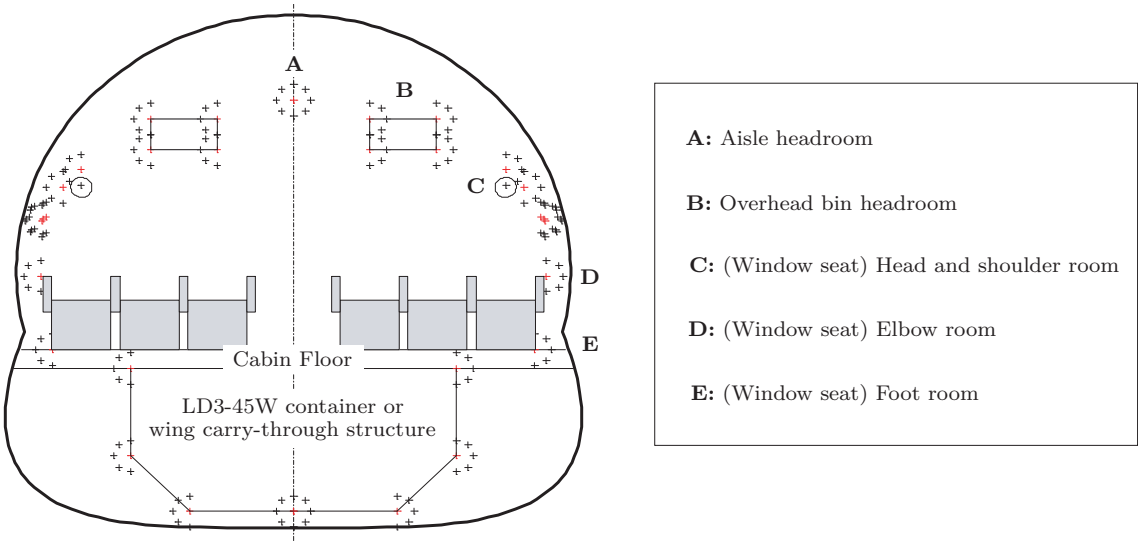


Figure 7.30: Cabin space constraint points (denoted by + symbols) related to passenger space requirements based on the size of a 95 percentile US male, as well as on the space needed for a fuselage frame of constant depth. The cargo lobe must accommodate the standard single-aisle container and/or the wing carry-through structure. A feasible cross-section shape is one that envelops all + symbols.

where each evaluation of $f_1(\vec{x})$ requires a Nelder and Mead pattern search to satisfy the constraint. The second objective f_2 is a measure of passenger comfort, formulated as:

$$f_2(\mathbf{x}) = \max\left[0.6W_{\mathrm{S}} + 0.1W_{\mathrm{A}} + 0.1W_{\mathrm{AR}} + 0.2H_{\mathrm{B}}\right]. \tag{7.16}$$

The third objective relates to the stress in the cabin structure. There are a variety of loads (e.g. bending loads, aerodynamic loads, etc.) which are dynamically applied to an aircraft fuselage in flight. Furthermore, the stress is non-uniformly distributed due to deviations in frame depth and circumferential size along various stations. A simplified stress analysis is proposed, which provides a fairly sensible measure of the harshness of the stress-state of a candidate design. A finite element analysis is still required, which is a physics-based, high-fidelity analysis. Distributed loads are applied to the frame circumference and the floor to signify pressurization loads and weight loads respectively.

A two-dimensional (linear static) finite element model is constructed using MSC Nastran, Inc. Numerical loading values are obtained from Niu [2002] as 0.35 psi and 8.25 psi for the floor and frame, respectively. The I-beam and C-section are constructed with a series of beam elements of depth 0.125 m. The material is aluminum Al2024-T3 with modulus of elasticity $E = 7.5 \times 10^{10}$ Pa and Poisson ratio of $\nu = 0.33$. The model is analyzed to determine the von Mises stress (σ_v) distribution around the frame. The simulation is completed by identifying the peak stress as the objective function, resulting

Variable	Definition	Lower bound	Upper bound
R^{CAR}	See Equation (7.12)	0.3	0.5
N_1^{CAR}	See Equation (7.12)	0.1	0.5
N_2^{CAR}	See Equation (7.12)	0.1	0.5
C^{CAR}	See Equation (7.12)	0.5	1.2
R^{PAX}	See Equation (7.13)	0.3	0.5
N_1^{PAX}	See Equation (7.13)	0.1	0.5
N_2^{PAX}	See Equation (7.13)	0.1	0.5
C^{PAX}	See Equation (7.13)	0.5	1.2
W_{S}	Seat width	0	1
W_{A}	Aisle width	0	1
W_{AR}	Armrest width	0	1
H_{B}	Bin headroom	0	1

Table 7.10: Design variable ranges for the fuselage cross-sectional design

Fuselage	f_1 (m^2)	f_2	f_3 ($\times 10^9$ Pa)
Reference	6.9928	0.5	1.4908

Table 7.11: Reference values for the Boeing 737 representative fuselage

in another first-order discontinuity. By minimizing the identified peak stress (where the location varies dependent on the cross-sectional shape), emphasis is indirectly placed on obtaining designs which exhibit a uniform stress distribution (i.e. circular cross-sections).

$$f_3(\mathbf{x}) = \min\{\max[\sigma_v(y(z))]\} \tag{7.17}$$

To reiterate on the concept of using KUPMOPSO to focus on and exploit the preferred compromise, the reference point is selected as the best approximation[2] of the Boeing 737 fuselage cross-section (see Figure 7.31 with the corresponding reference values given in Table 7.11). This cross-section has a slight hint of a double-lobe design where the cusp point coincides with the floor. The quasi-circular cross-section implies a fairly uniform circumferential stress distribution whilst maintaining a relatively low cross-sectional area for reduced pressure drag. The solution spread is specified as $\delta = 5 \times 10^{-2}$.

The allowable computational budget for this problem is directly related to the multi-modality of the objective landscapes. The first-order discontinuity of the objectives places further limitations on the use of Kriging models, since a larger number of sample points

[2]This case study is merely a demonstration of the algorithm on a realistic design problem. Only the key objectives and constraints are taken into account, hence actual results may be different from those obtained if the same exercise was conducted in an industrial setting

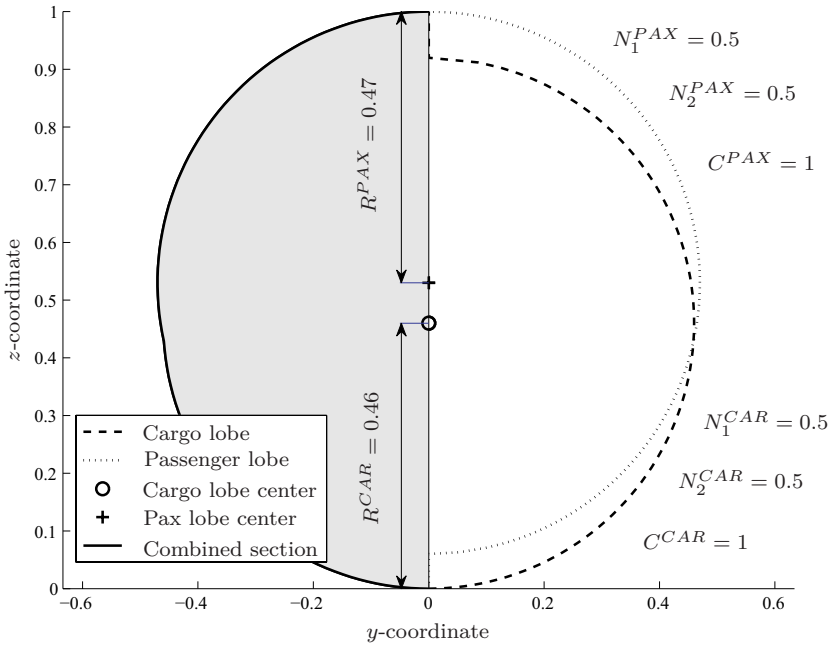


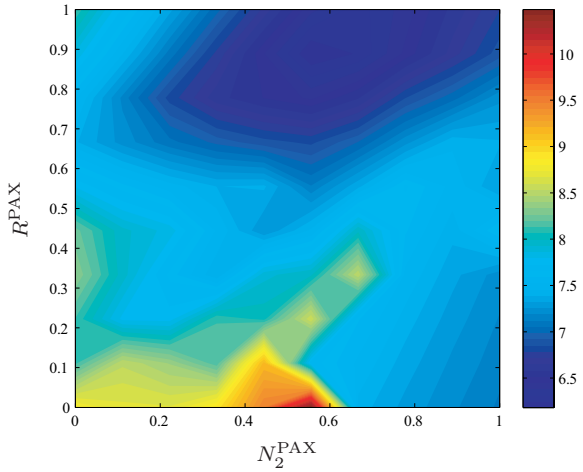
Figure 7.31: Reference point Boeing 737 representative geometry (normalized co-ordinates)

are required to accurately depict non-smooth regions of the design space. Visualizing the design landscape (see Section 4.4.1) is a useful and simple technique to gain insight into a discontinuous landscape. Selected contour slices of the multi-dimensional design space are shown in Figure 7.32. The deceptive landscapes originate from the element of uncertainty in the location of the peak stress for each individual cross-sectional design, as well as the requirement of the cross-section to enclose all constraint points. In a preference-based framework, the Kriging predictions are localized entirely within the preferred region of the design space, thereby alleviating the irregularity of the landscape.

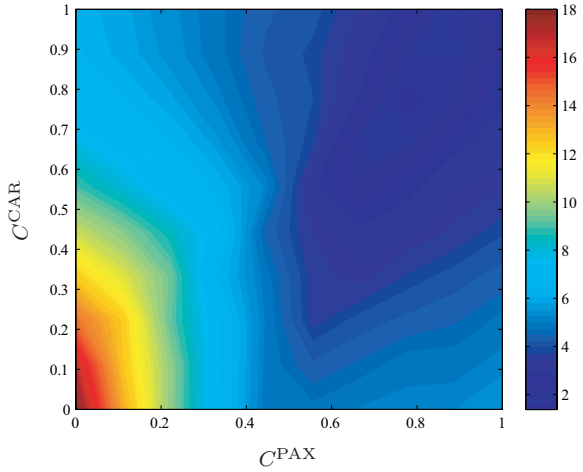
Optimization Results

A swarm population of $N = 100$ particles is initialized. A Kriging model based on an LHS design of $N_K = 100$ points is constructed for objectives f_1 and f_3. Figure 7.33 illustrates the CV error of objective f_3. The accuracy of the global Kriging models is fairly low here as a result of modelling discontinuities. However, since a user-preference module is adopted, the Kriging models will focus entirely within the preferred region, providing a more accurate depiction of the localized area. A computational budget of 450 precise evaluations is imposed, due to the first-order discontinuities.

A further 350 precise evaluations of the objective array were performed over $t \approx 170$ time-steps before the evaluation limit was breached. The additional 350 evaluations are composed of 280 particles and 70 mutations which satisfied the reference point screening



(a) Coloured by area (m^2)



(b) Coloured by peak stress ($\times 10^9$ Pa)

Figure 7.32: Selected contour slices for various parameters (normalized variable ranges)

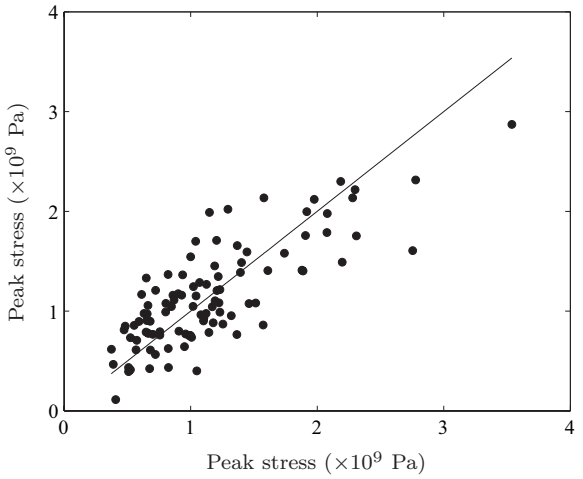
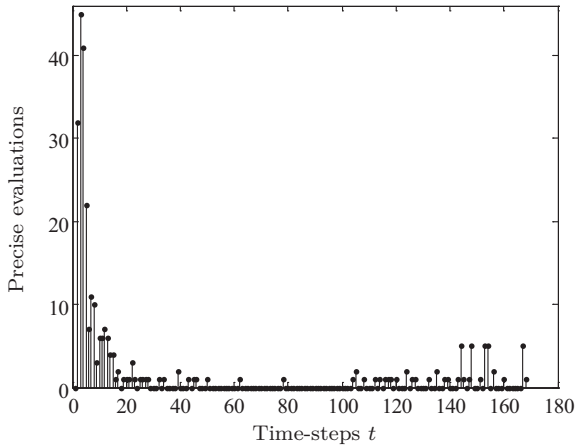
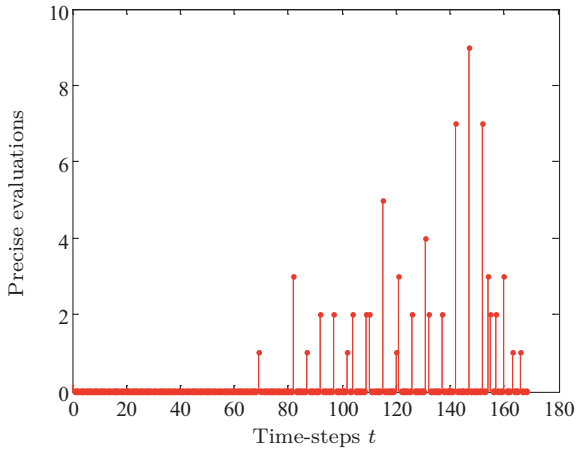


Figure 7.33: CV error of Kriging training dataset for f_3 ($CV(\mathbf{Y}) = 11.34\%$)



(a) Evaluations due to updates

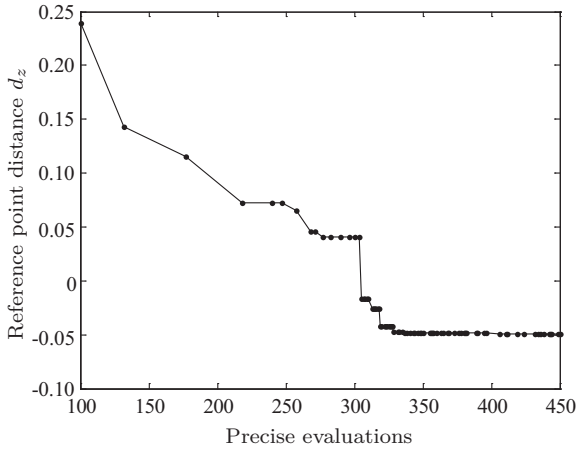


(b) Evaluations due to successful mutations

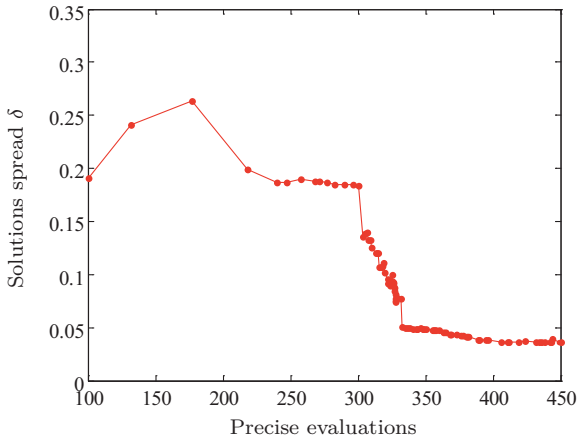
Figure 7.34: History of precise evaluations for the fuselage cross-sectional design case-study

criterion, as shown in Figure 7.34. It is observed from the stem plot in Figure 7.34(a) that the largest number of update points occur in the earlier stages of the search. This is due to the relatively poor prediction accuracy of the global Kriging models. As the training datasets of the Kriging models become localized within the identified preferred region, a reduced percentage of particles satisfy the reference point screening criteria. The mutation stem plot shown in Fig. 7.34(b) conversely illustrates that during the initial phases of the search, the searching prowess of the swarm is sufficient to gain enough diversity and thus mutations are not recorded. As the search stagnates and there is less consistent improvement in the archive, particles are scheduled for mutation in an attempt to encourage further improvement. This ensures that the mutation operator has been used effectively, as it does not disrupt the inherent searching capabilities of the swarm.

Figure 7.35(a) features the progress of the preferred design $\mathbf{z}'$. For this simulation, 100 precise evaluations are required to reach within 65% of the final converged design. After



(a) Progress of the most preferred solution

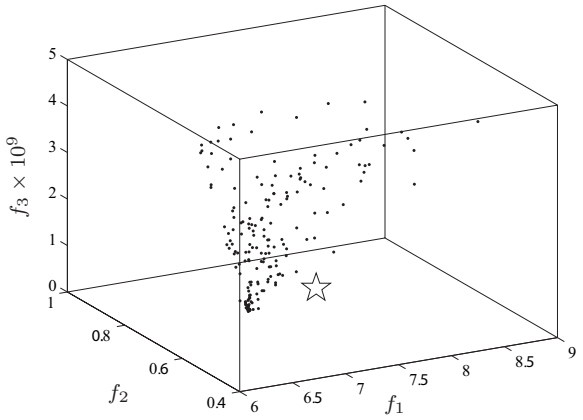


(b) History of spread of solutions

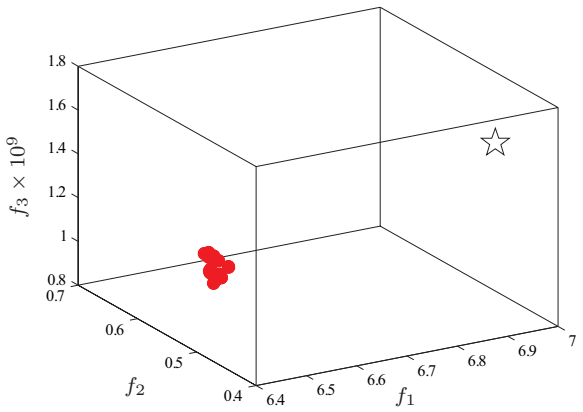
Figure 7.35: Convergence performance for the fuselage cross-sectional design case-study

approximately 335 evaluations, the search has reached within 3% and appears to have converged since from this point on only slight reductions in $\mathbf{z}'$ are observed as the swarm further exploits the preferred region. Figure 7.35(b) alternatively features the history of the solution spread δ. Although the algorithm does not obtain the desired spread of 0.05, a fairly consistent spread of $\delta \approx 0.04$ is observed after 330 evaluations.

The performance of KUPMOPSO is further demonstrated in Figure 7.36(a) which features the 200 most recent precisely evaluated solutions. Attraction towards the preferred region is observed, which progressively becomes localized. Furthermore, few solutions appear to disturb the search direction of the algorithm. The reference point criterion is also shown to function appropriately with the mutation operator, which is non-destructive. Featured in Figure 7.36(b) is the final set of 23 designs. The KUPMOPSO algorithm has identified solutions which exhibit improvement over all reference values.



(a) Most recent precise evaluations



(b) Identified non-dominated solutions

Figure 7.36: Precise evaluations performed and the resulting non-dominated solutions

Final Designs

The percentage improvement of the identified preferred design over each objective with respect to the reference point geometry is shown in Table 7.12. The geometry of the preferred design is featured in Figure 7.37. It is observed that the cusp point is no longer as evident as the reference design. The preferred geometry follows a more circular form, which corresponds to the 33% recorded improvement in the stress. An evident reduction in area is visible from the passenger lobe as well as a more comfortable seating arrangement for passengers due to the elongation in the width. The circumferential stress distribution

Fuselage	f_1	f_2	f_3
Reference	6.9928	0.5	1.4908
Preferred	6.4952	0.552	0.9915
% Improvement	7.1	10.4	33.5

Table 7.12: Objective values for the preferred and reference designs with respective improvement

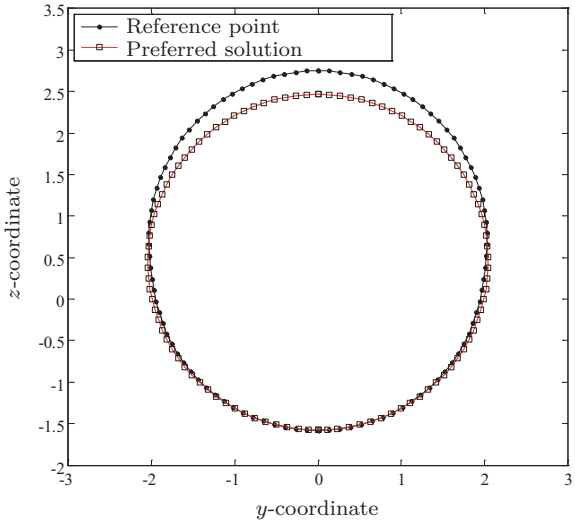


Figure 7.37: Geometry of the preferred design compared with the Boeing 737-style design

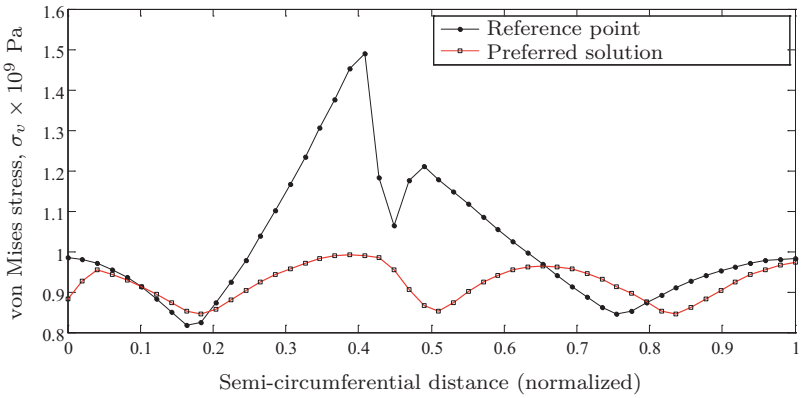
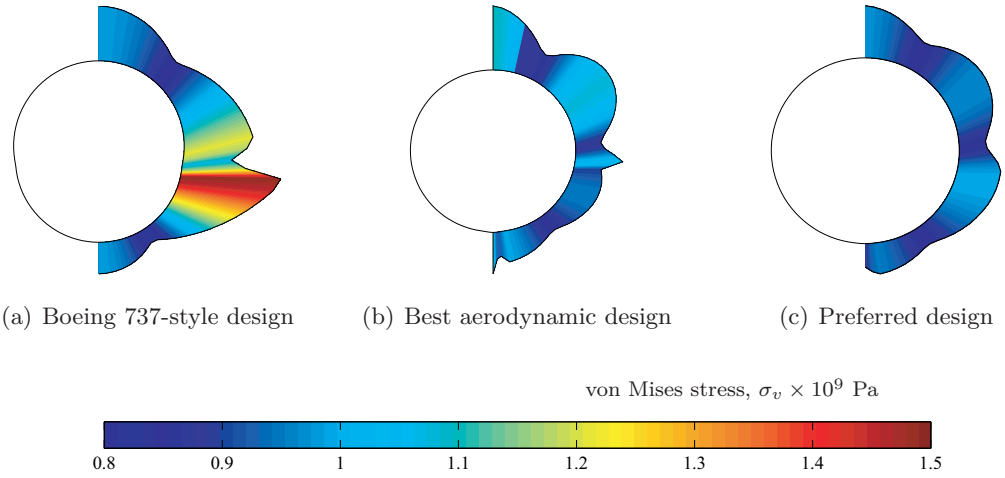


Figure 7.38: The circumferential stress distribution over the preferred and reference designs

around the reference and preferred geometries is featured in Figure 7.38. The von Mises stress is recorded at each circumferential station which is normalized with respect to the z-axis. In addition to a reduced overall operating stress, the stress distribution of the preferred geometry is very uniform compared with the reference design. The peak stress for both geometries occurs at the intersection of the floor.

Figure 7.39 introduces the design which provides additional improvements of 0.25% and 0.2% respectively in area and comfort, at the expense of a 9.5% increase in the peak stress. This is the solution with the best recorded area and thus the most aerodynamically proficient design. The associated stress contours are illustrated. For the best aerodynamic design an evident increase in the peak stress, which is located at the floor intersection, is observed over the preferred design.



(a) Boeing 737-style design (b) Best aerodynamic design (c) Preferred design

Figure 7.39: Precise evaluations performed and the resulting non-dominated solutions

7.5 Aeroacoustic Optimization of Trailing-Edge Flow

In this section, a noise reduction problem for laminar unsteady trailing-edge flow is considered. Nearly all vehicles moving through a fluid produce noise (e.g. airframe noise, hydrofoils, rotor blades, wind turbines, etc.). Hydrofoil noise is an area of great interest for military applications, particularly for submarine detection and design. In naval applications, hydrofoils that generate lift are used for both propulsion and control surfaces, requiring that they must meet specific performance criteria on lift and drag. For military applications, hydrofoils must also be designed to meet noise and detection requirements. Blake [1986] demonstrates that relatively small changes in the geometry can lead to substantial changes in the aeroacoustic performance. Marsden *et al.* [2004] performed computational experiments for laminar trailing-edge flow and established the dependence of the acoustic energy generated due to shape modifications, using their surrogate management framework. This work was later confirmed by Rumpfkeil and Zingg [2008]. In both cases, significant reductions in the radiated acoustic power were achieved.

Since computing the acoustic source terms requires a time-accurate solution to the Navier-Stokes equations, the construction of a surrogate model is essential. The airfoil geometry is a shortened version of the airfoil used in the experiments by Blake [1975], as shown in Figure 7.40. The airfoil is similar to that used by Marsden *et al.* [2004], and Rumpfkeil and Zingg [2008]. The airfoil chord is ten times its thickness with a semi-circular leading edge and beveled trailing edge. The freestream Mach number is $M_\infty = 0.2$ with a Reynolds number of $Re = 10^4$ and an incidence angle $\alpha = 0°$. The highlighted section denotes the region which is allowed to deform during the optimization process.

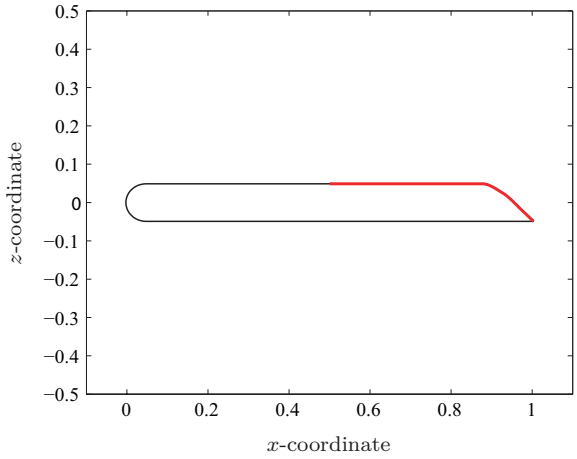


Figure 7.40: Blake airfoil geometry used for the aeroacoustic optimization case-study

Acoustic Analogy

Although acoustic pressure fluctuations are by definition a compressible flow phenomenon, at low Mach numbers the acoustic source terms can be approximated by an incompressible flow solution, and then coupled with an acoustic analogy to compute the far-field noise. The pioneering work of Lighthill [1952] provides a theoretical framework for decoupling the source field computation from noise propagation computations. Significant work has been done by Wang *et al.* [1996] towards the development of numerical techniques for trailing-edge aeroacoustic computations using Lighthill's analogy. In this work, a model problem of low Reynolds number unsteady laminar flow over a symmetrical airfoil is used to quantify the influence of surface dipoles and volume quadrupoles using Curle's extension to Lighthill's analogy [Curle, 1955]. The numerical results confirm the quantitative dominance of surface pressure dipoles to far-field sound at low Mach numbers, so that Curle's formulation for an acoustically compact surface is reduced to,

$$p(\mathbf{x},t) \approx \frac{x_i}{4\pi c_0 \left|\mathbf{x}\right|^2} \cdot \frac{\partial}{\partial \tau} \oint_S (n_j p_{ij})\, \mathrm{d}S, \tag{7.18}$$

where $\tau = t - \mathbf{x}/c_0$ is the retarded time to the receiver $\mathbf{x}$, c_0 is the speed of sound, p_{ij} is the compressive stress tensor and n_j is the unit normal to the airfoil surface S. For a complete derivation, refer to Howe [1998]. An objective function can thus be defined, that is directly proportional to the radiated acoustic power per unit span,

$$f = \overline{\left(\frac{\partial}{\partial t} F_x\right)^2} + \overline{\left(\frac{\partial}{\partial t} F_y\right)^2}, \tag{7.19}$$

where $F_i = \oint_S (n_j p_{ij})$ is the net force exerted on the fluid by the acoustic surface in the i^{th} direction. The overbar denotes time-averaging for the time-interval. The acoustic radiation is of dipole type, caused by the fluctuating lift and drag forces.

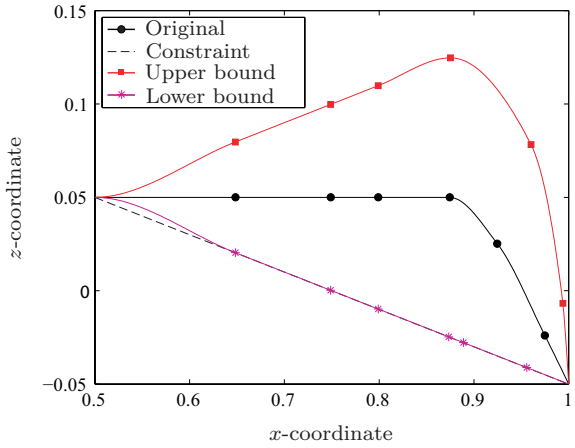


Figure 7.41: Normal displacements of the spline control points measured from the original airfoil

Trailing-Edge Shape Parameterization

The right half of the upper surface of the airfoil is allowed to deform during the optimization process. The airfoil surface is defined by interpolation through a finite number of control points using a Hermite spline, and displacement of the airfoil surface is normal to the surface tangent of the original airfoil geometry [see Marsden *et al.*, 2004]. Displacement of the control points is normalized to the maximum allowable normal distance of the airfoil surface. Each control point a_i has values $-1 \leq a_i \leq 1$, such that $a_i = 0$ corresponds to the original airfoil surface. Use of a Hermite spline as the interpolating function guarantees that values of all interpolated points are bounded by the minimum and maximum values of the known control points. This ensures that no point on the surface will be displaced further than the maximum allowable distance. A total of six control points are used to define the trailing edge geometry. The minimum thickness requirement defined by Marsden *et al.* [2004] is imposed, which is given by a straight line connecting the left edge of the deformation region and the trailing edge, as shown in Figure 7.41. The normal displacement is governed by the thickness constraint and is given as,

$$|\mathbf{r_i}| = a_i \cdot \left[(x_i - x_L)^2 + (y_i - y_L)^2\right]^{\frac{1}{2}} \tag{7.20}$$

where (x_i, y_i) is the original surface point and (x_L, y_L) refers to the intersection point of the line normal to the airfoil surface and the line defining the minimum thickness.

Flow Solver

The two-dimensional Navier-Stokes equations are solved using ANSYS Fluent with a second-order implicit time marching formulation. A C-mesh of 450×90 nodes stretching 15 chord lengths aft of the trailing edge is generated for each candidate geometry. This

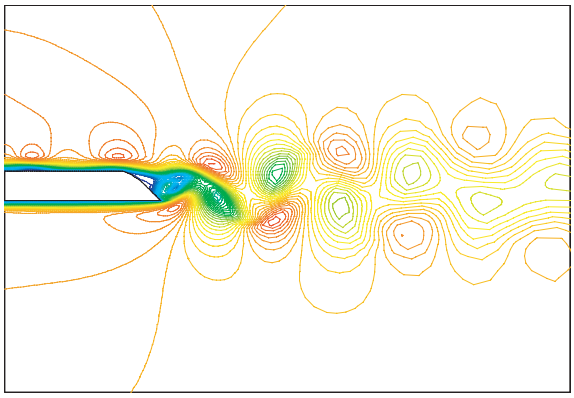


Figure 7.42: Contours of velocity magnitude illustrating unsteady vortex shedding

mesh size provides a reasonable balance between solution accuracy and computational expense. The pressure-based solver with second-order spatial discretization is selected. The time is made dimensionless by a factor of c/U_∞ where c is the airfoil chord length and U_∞ is the freestream velocity. The laminar flow around the Blake airfoil exhibits unsteady vortex shedding, as shown in Figure 7.42.

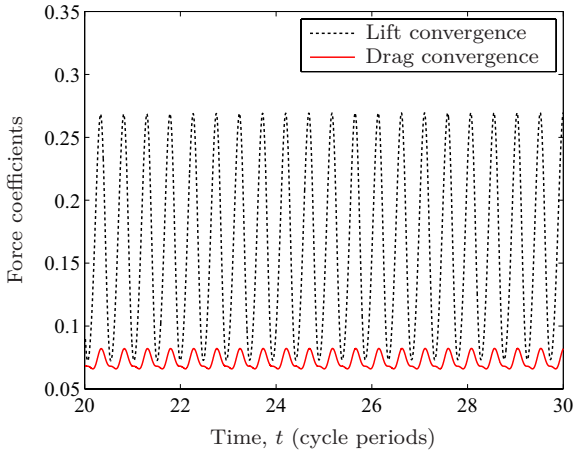
The vortex shedding shown in Figure. 7.42 yields oscillating force coefficients over time as illustrated in Figure 7.43(a). As a result, the acoustic radiation calculated by Eq. (7.19) also oscillates with time. The simulation is therefore performed until the solution achieves a quasi-steady state. Time averaging is then carried out for a sufficiently long time until Eq. (7.19) has converged. To achieve a quasi-steady state, the simulation is first performed until $t = 20$ (cycle periods), and then time-averaging is subsequently carried out until $t = 40$. The non-dimensional time-step is selected as $\Delta t = 0.002$. Figure 7.43(b) shows the convergence history of Eq. (7.19) compared with the oscillatory function. The objective function is scaled by the freestream Mach number M_∞ for comparison with the results published in Marsden *et al.* [2004] and Rumpfkeil and Zingg [2008]. The agreement with these results is favourable, confirming the grid density and time-resolution is sufficient.

Problem Formulation and Reference Point

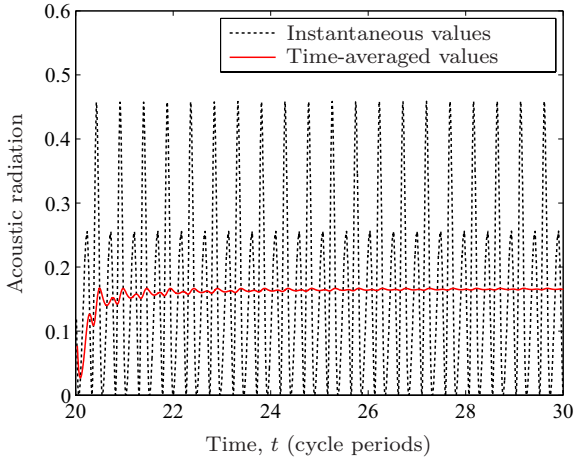
The studies of Marsden *et al.* [2004] and Rumpfkeil and Zingg [2008] were limited to a single-objective problem to reduce the radiated acoustic power. In this study the problem is extended to combine two conflicting objectives. The thickness constraint is regulated implicitly by the parameterization model. The two objectives are represented as,

$$f_1 = \min \overline{\frac{1}{1+F_y}}, \tag{7.21}$$

$$f_2 = \min \overline{\left(\frac{\partial}{\partial t}F_x\right)^2} + \overline{\left(\frac{\partial}{\partial t}F_y\right)^2}. \tag{7.22}$$



(a) Time-history of force

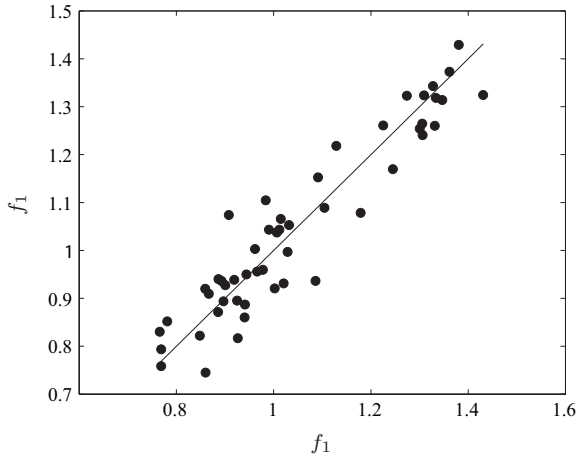


(b) Time-history of Eq. (7.19)

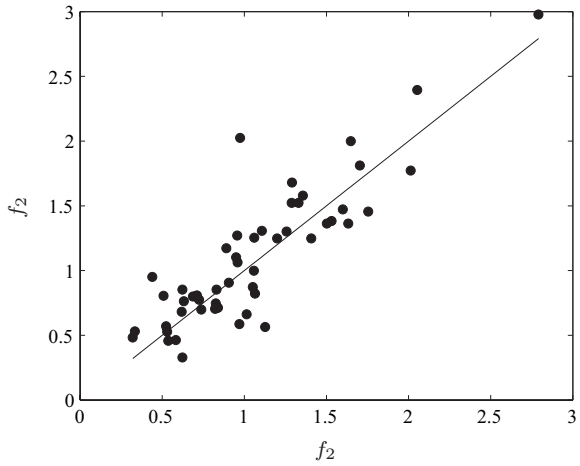
Figure 7.43: Convergence history for the original Blake airfoil (oscillation due to vortex shedding)

The objective functions make no reference to minimizing the drag coefficient as it was established in Marsden *et al.* [2004] that improving the aeroacoustic performance led to designs exhibiting reduced drag characteristics. However the results demonstrated that aeroacoustic performance was only linked to the magnitude of lift fluctuation, rather than the average lift value. The first objective f_1 therefore relates to maximizing the time-averaged lift generated by the profile to maximize performance during forward motion. The second objective utilizes the acoustic analogy introduced earlier to reduce noise and detection.

The reference point is selected as the original Blake geometry (see Figure 7.43) in order to identify designs which maintain a similar compromise between the lift generated and the radiated acoustic power.



(a) Training sample for f_1 ($CV(\mathbf{Y_1}) = 9.59\%$)



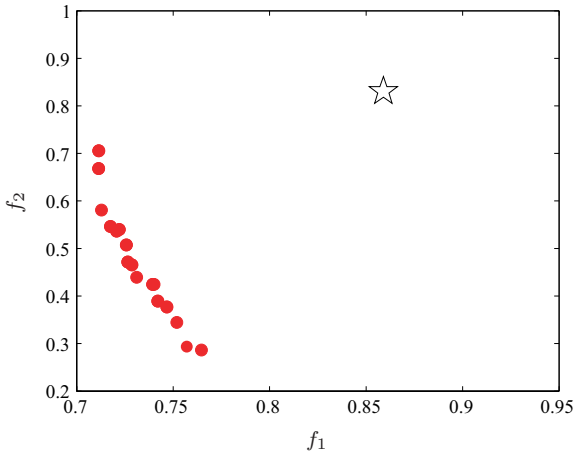
(b) Training sample for f_1 ($CV(\mathbf{Y_2}) = 10.79\%$)

Figure 7.44: CV plots for the constructed Kriging models

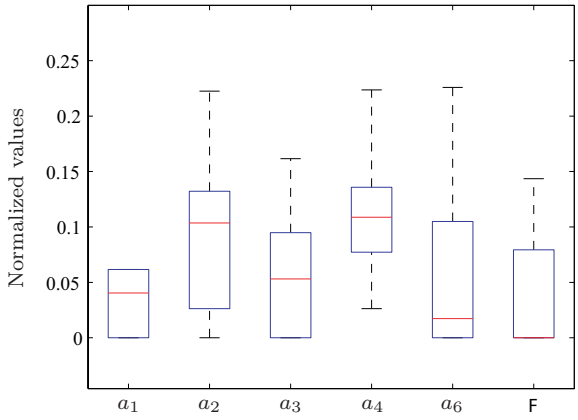
Optimization Results

Kriging models are constructed based on an LHS design of $N_K = 50$ samples. A limited sample size has been selected to construct the global models, which allows for more precise evaluations to be allocated during the search which is advantageous to maintain a high prediction accuracy within a localized area of the design space. Figure 7.44 features the CV plots for the constructed Kriging models of f_1 and f_2. The high CV error is not of immediate concern at this stage, and it is expected that the prediction accuracy will progressively improve as the preferred region becomes more localized.

A swarm population of $N = 100$ particles is initialized and flown until the imposed computational budget of 200 precise evaluations is breached. The solution spread was controlled using $\delta = 1\times10^{-2}$. For every precise evaluation, the mesh was partitioned across 12 cores of the VPAC supercomputing cluster with AMD Barcelona 2.3 GHz quad-core



(a) Identified non-dominated solution set



(b) Normalized box-plots of variable ranges

Figure 7.45: Results of the aeroacoustic trailing edge design case-study

processors. Each simulation required approximately 3 hours to complete. An additional 150 precise evaluations were performed during the search. This resulted in a final set of 30 non-dominated designs which are featured in Figure 7.45(a).

The identified preferred designs can be further visualized through the box-plot shown in Figure 7.45(b). Variable values are normalized hence the lower bound of the graph is representative of the maximum negative displacement of each control point a_i. It is observed that the preferred designs have their control points positioned sinusoidally along the trailing edge. This sinusoidal or *wavy* trailing edge geometry was also observed in the studies of Marsden *et al.* [2004]. All designs have a collective negative displacement with reference to the reference point trailing edge.

Design	f_1	f_2
Reference point	0.8590	0.8306
Preferred	0.7310	0.4396
min f_1	0.7114	0.6685
min f_2	0.7769	0.2989

Table 7.13: Results of the aeroacoustic trailing edge design case-study

Final Designs

The catalogued preferred designs of this case-study are the most preferred design, the best aerodynamic design (i.e. min f_1) and the best aeroacoustic design (i.e. min f_2). Table 7.13 shows the numerical results of the identified designs of interest. The preferred design offers a 15% and 47% improvement for objectives f_1 and f_2 over the reference point, respectively. The best aerodynamic design offers a further 2% improvement in the maximum lift generated at the expense of only a 19% reduction in the noise generated. Marsden *et al.* [2004] documents a 90% reduction in the total radiated acoustic power with no lift improvement (this study utilized fifteen B-spline control points to parameterize the trailing edge). In the current study, the best aeroacoustic design provides a 65% reduction in the radiated acoustic power with the added advantage of a 10% increase in lift.

The trailing edge geometries of the identified designs of interest are featured in Figure 7.46. The sinusoidal geometry of each preferred design variant is clearly visible. It is observed that the extent of this sinusoidal geometry is relative to the reduction of the total radiated acoustic power. It is therefore concluded that the waves serve to suppress the generation of trailing edge vortices (reducing the fluctuation in the force coefficients) thereby reducing the generated noise. The aerodynamic design clearly aims to provide maximum aft camber to generate lift, whilst maintaining enough sinusoidal deviation in the trailing edge to suppress the generation of vortices.

Figure 7.47 illustrates the surrounding flow-field of the identified designs of interest. The disturbance in the wake is a measure of the vortex generation. It is clearly shown that the reference point geometry (see Figure 7.47(a)) sheds a consistent array of vortices, propagating from the trailing edge. Alternatively the preferred designs (especially the aeroacoustic design) are shown to shed vortices non-uniformly. The disturbance in the wake of the preferred designs is not as large, which is a result of suppressed vortices, as well as a higher dissipation rate.

The non-uniformity of the vortex shedding is a result which has not been documented before for this case-study (to the best knowledge of the author). The generation of vor-

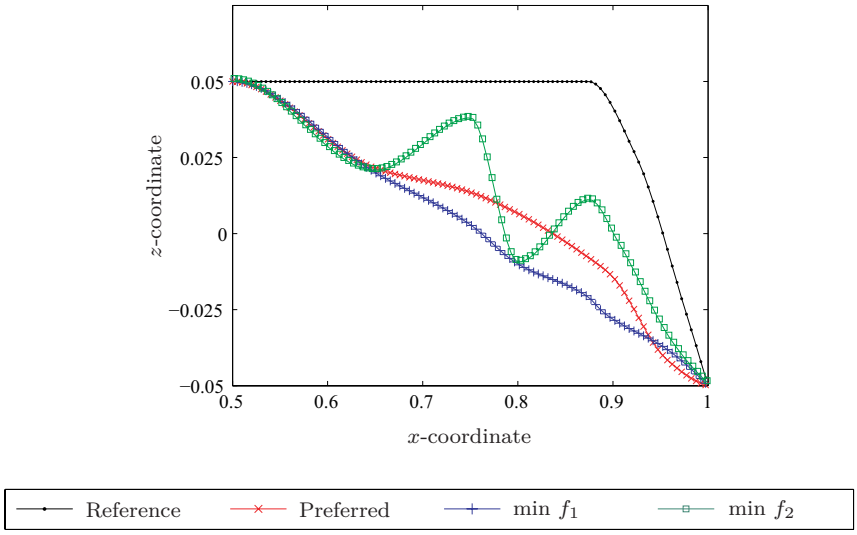
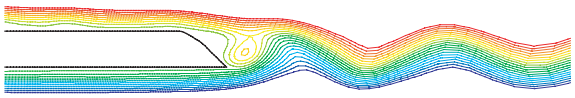
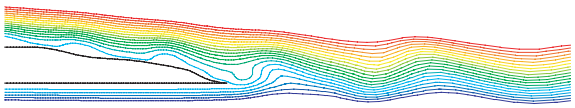


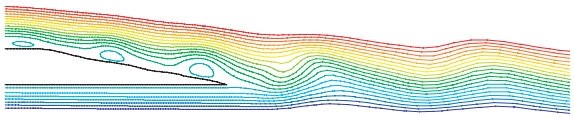
Figure 7.46: Trailing-edge geometry of the identified designs of interest



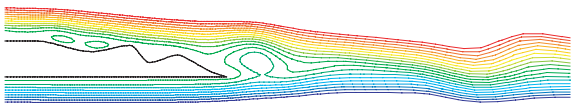
(a) Reference point design



(b) Preferred design



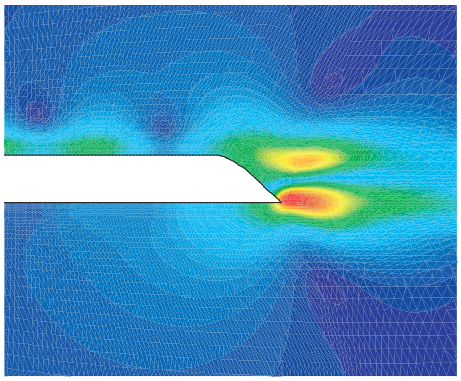
(c) Best aerodynamic design



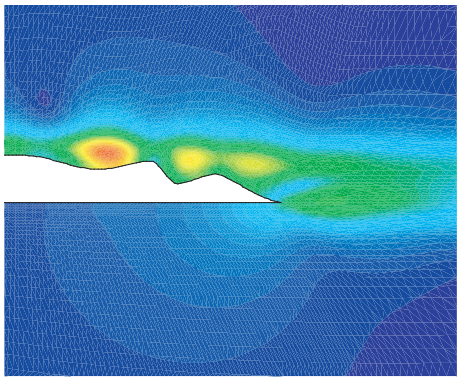
(d) Best aeroacoustic design

Figure 7.47: Contours of stream function illustrating the flow-field

tices can be more clearly visualized using statistical time-averaged data collected from the simulation. Figure 7.48 features contours of RMS (Root-Mean-Square) static pressure. Areas of high RMS pressure provide a fairly clear observation of where vortices are forming. Figure 7.48(a) features the reference point design, and it is evident that vortices are shed fairly uniformly from the pressure and suction surface which yields the oscillating force coefficients over time. Alternatively the best aeroacoustic design (with the charac-



(a) Reference point design

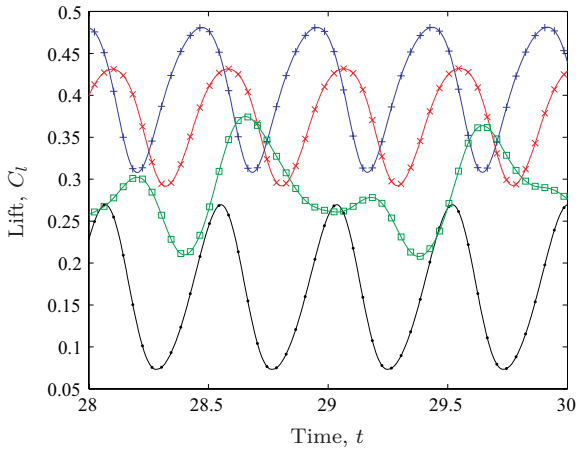


(b) Best aeroacoustic design

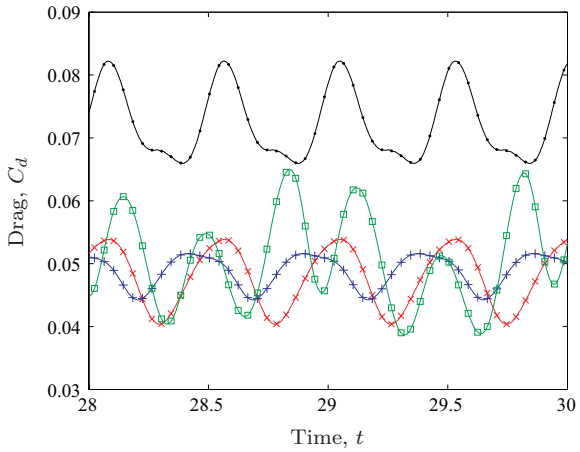
Figure 7.48: Contours of RMS static pressure

teristic sinusoidal trailing edge) clearly shows the formation of other vortices due to the wavy geometry. The magnitude of the vortex strength progressively reduces yielding less disturbance at the trailing edge.

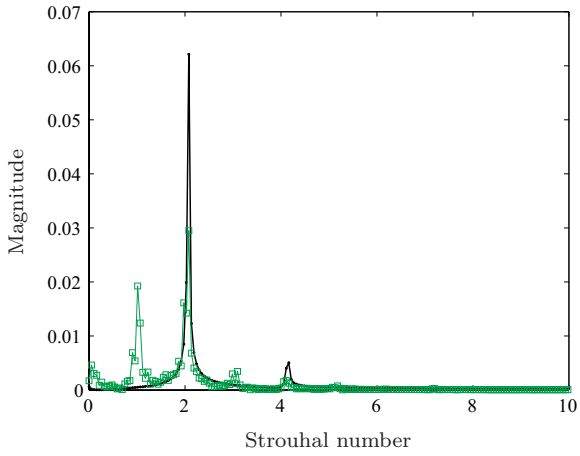
To gain further insight to the physical reasoning behind this phenomenon it is necessary to visualize the force fluctuation. The formation of additional vortices due to the trailing edge geometry will result in changes to the oscillation of the force coefficients over time. Figures 7.49(a) and 7.49(b) illustrate a segment of the time-history convergence for the lift and drag coefficients of the identified preferred designs. The lift and drag coefficients of the reference point design oscillate in phase, with an additional mid-phase vortex shed due to the fluctuating drag coefficient. This is evident from Figure 7.49(c) which clearly shows two peaks in vortex shedding, at Strouhal numbers of approximately 2 and 4. The coefficients of the preferred and best aerodynamic designs oscillate at a lower amplitude, and the vortex shedding is in fact shown to be similar (yet out of phase) to the reference point design. Alternatively the best aeroacoustic design oscillates at various frequencies confirming the formation of several phases of harmonic fluctuation (as per Figure 7.48(b)).



(a) Instantaneous lift values



(b) Instantaneous drag values



(c) Vortex shedding due to lift fluctuation

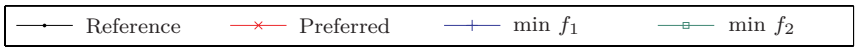


Figure 7.49: Performance comparison of each of the designs of interest

Figure 7.49(c) further demonstrates the appearance of other fluctuations, since minor peaks are now evident at Strouhal numbers of approximately 1, 3 and 5.

7.6 Aerodynamic High-lift Configuration Design

This section describes the final application of this framework for the aerodynamic design of a transport high-lift configuration. To achieve sufficient low-speed performance without compromising cruise performance requires a fairly sophisticated high lift configuration [van Dam, 2002]. The primary goal of an aerodynamic high-lift system is to increase payload-carrying capacity and reduce take-off and landing distances by maximizing the lift coefficient for a given angle of incidence, without the onset of massive flow separation [Smith, 1975]. Studies confirm that relatively small changes in the aerodynamic performance of a high-lift system potentially translate to major benefits in aircraft payload-carrying capacity and performance [Garner *et al.*, 1991]. For this reason, high-lift aerodynamic and system design remain at the forefront of aerospace research.

For a typical high-lift configuration design, involving a forward (*slat*) element and an aft (*flap*) element, the flow field is physically complex due to the element interactions [Smith, 1975; van Dam, 2002]. Combined with the intricacy of system support and actuation, this has traditionally led to an experiment intensive development process. However, due to the recent development of numerical modelling capabilities, CFD techniques are superseding physical experimentation design methods [Rumsey and Ying, 2002]. In particular, the significant progress achieved in synthesizing automated optimization tools and high-fidelity CFD has significantly reformed the aerodynamic high-lift design philosophy [Kim *et al.*, 2004; Keane and Nair, 2005].

Problem Formulation

High-lift configuration design is a highly multi-disciplinary process. For the present study, the problem is simplified to a two-dimensional aerodynamic optimization of a high-lift configuration. This simplification allows for a higher-fidelity solver, providing an interesting challenge for the developed algorithm. The baseline three-element configuration for this optimization study is selected as the McDonnell-Douglas 30P/30N, shown in Figure 7.50. The 30P/30N configuration has already been highly optimized for maximum lift. Wind-tunnel measurements have been extensively performed for this configuration [Ying *et al.*, 1999], and the results of many CFD computations for this geometry have been reported using a variety of numerical schemes [Rogers *et al.*, 1994; Rumsey *et al.*, 1998].

The variables used in the optimization study are the relative positions of the slat and

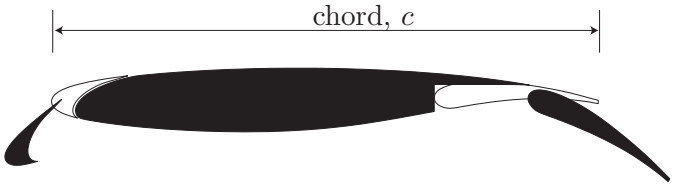


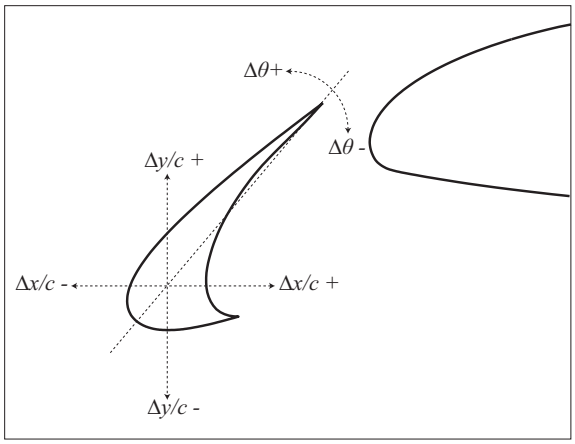
Figure 7.50: The MDC 30P/30N configuration with deployed slat and flap components

flap components. These variables measure the gap, overlap and deflection angle relative to the main component. For the present study, relative positions of the slat and flap are specified indirectly, through $\pm\, x/c$ and $\pm\, y/c$ directional translations, and rotation $\pm\, \theta$. In these measurements, c refers to the clean airfoil chord with retracted flap and slat (as shown in Figure 7.50). In this way, the design variables are geometrically independent and their relative influence on the design space is easily identifiable. The design variables are initialized relative to the position of the original configuration. Figure 7.51 illustrates the translational and rotational displacement of the slat and flap components.

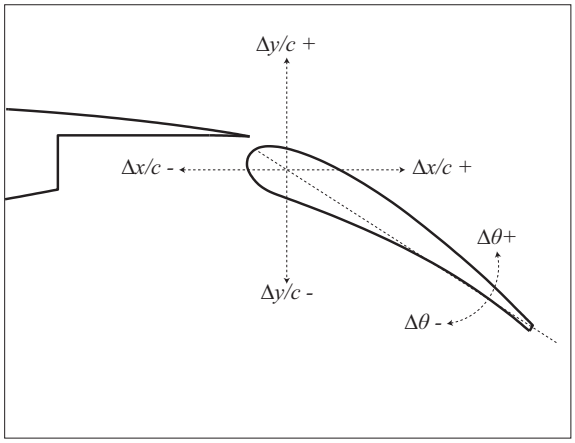
The proficiency of the proposed framework is demonstrated by maximizing the lift generated on the 30P/30N configuration at various design conditions, motivated by the work of Ying *et al.* [1999]. All simulations are performed at a Reynolds number $Re = 9 \times 10^6$ and Mach number $M = 0.2$. The reference point geometry is logically selected as the original 30P/30N configuration. Three objectives are formulated for the problem, which are $f_1 = \min(1/C_l)$ and $f_2 = \min(C_d/C_l)$ at $\alpha = 8°$, and $f_3 = \min(1/C_l)$ at $\alpha = 19°$, where C_l and C_d are the configuration lift and drag coefficient respectively. The objectives f_1 and f_2 are optimized at an angle of incidence $\alpha = 8°$, typical of an approach configuration. The objective f_3 is optimized at an angle $\alpha = 19°$, nearing the angle where maximum lift is generated. Neither objective is discontinuous, but the third objective is highly non-linear, as some configurations will lead to premature separation and stall. The boundaries of the design space are selected such that there is a smooth transition of the grid and do not result in any infeasible configurations (e.g. component intersection, etc.). The design variable ranges are shown in Table 7.14.

Multi-block grid

Due to the complexity of the geometry, most methods for high-lift configurations are typically based on unstructured grids or overset grids [Vatsa *et al.*, 1994]. These strategies are optimal for complex geometries, yet the computational efficiency and accuracy is poor relative to an orthogonally structured mesh. In this study, the block-structured grid generator Gridpro is utilized. The grid topology is divided into a number of many-to-one



(a) Slat component parameterization

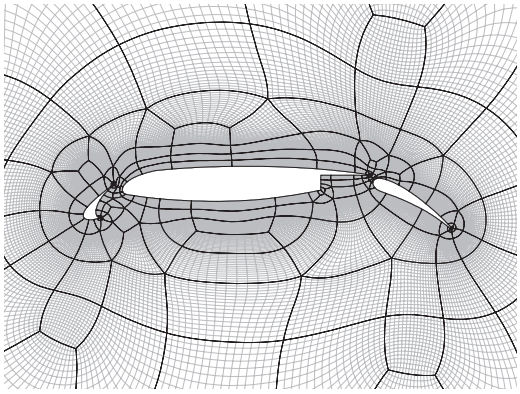


(b) Flap component parameterization

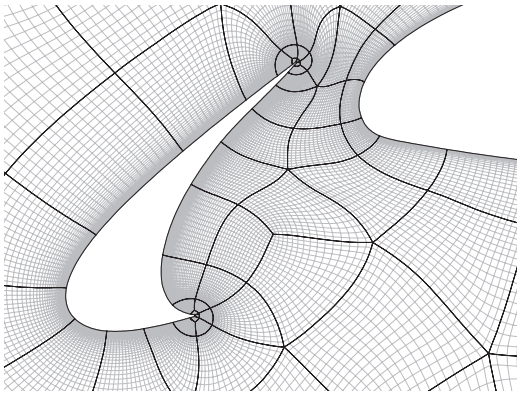
Figure 7.51: Parameterization variables for high-lift configuration

elementary blocks, which allows for accurate meshing of geometry, without the need to propagate dense grids to the computational far-field. Figure 7.52(a) illustrates a partial view of the grid topology generated for this problem.

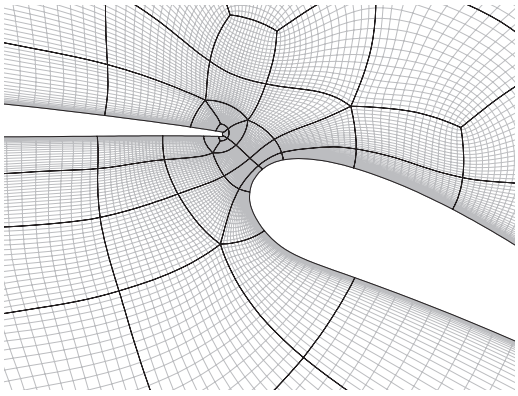
The O-grid topology consists of 234 blocks, resulting in a total mesh size of approximately 80000 elements. The far-field boundary extends $20c$ radially from the origin. The grid is algebraically clustered at the surface where the first grid point has a y-plus magnitude of $O(1)$, which is critical for adequate resolution of the boundary layer. It is shown in Figure 7.52(a) that there is a rather dense topology in regions of interest and the topology around the trailing edges of each component are mapped to an interior circular surface to preserve orthogonality. Also shown are close-ups of the slat and flap grids. Gridpro generates the resulting grid by individually meshing each topology block. Therefore, during the optimization process, the airfoil element surfaces and surrounding topological blocks are translated and rotated according to the design variable notation.



(a) Full multi-block grid topology



(b) Slat multi-block grid



(c) Flap multi-block grid

Figure 7.52: Multi-block grid topology generated for the 30P/30N configuration

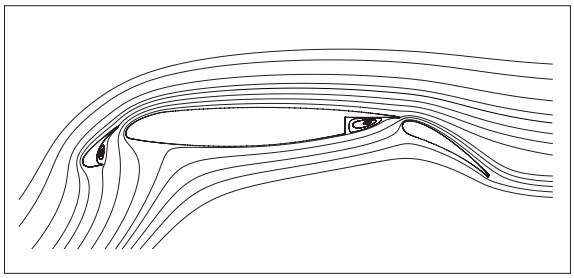
Parameter	Description	Lower bound	Upper bound
Δx_S	Slat translation $\Delta x/c$	-0.04	0.01
Δy_S	Slat translation $\Delta x/c$	-0.03	0.04
$\Delta \theta_S$	Slat rotation $\Delta\theta$	-10°	10°
Δx_F	Flap translation $\Delta x/c$	-0.02	0.04
Δy_F	Flap translation $\Delta x/c$	-0.07	0.005
$\Delta \theta_F$	Flap rotation $\Delta\theta$	-10°	10°

Table 7.14: Design variable ranges for optimization study

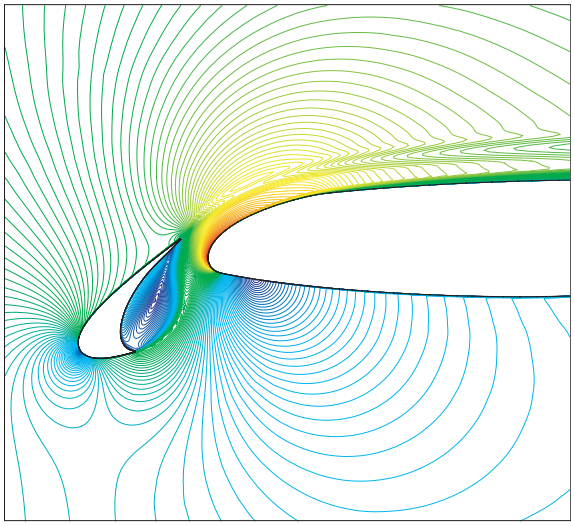
Flow solver

Significant progress in formulating a theoretical basis for high-lift aerodynamics was pioneered by the work of Smith [1975]. Figure 7.53(a) illustrates the stream pattern over the MDC 30P/30N configuration. The circulation of a forward element or the slat effect reduces the leading edge suction peak, thus delaying separation. The trailing element however induces a circulation effect on the forward element which tends to increase the loading or lift generated. These flow phenomena are predominantly inviscid, yet viscous effects also arise due to the individual wakes from each element. While wakes reduce the pressure peak of trailing elements, they often tend to merge with the boundary layer of the trailing element resulting in a thicker shear layer, termed a confluent boundary layer, increasing the likelihood of separation. Figure 7.53(b) shows the development of the confluent boundary layer as the wake from the slat element merges with the forming boundary layer of the main element.

To portray each of these flow phenomena, a high-fidelity flow solver is essential. In this study, the general purpose finite volume code ANSYS Fluent is used. The compressible steady-state RANS equations are solved with the implicit pressure-based scheme using second order upwind spatial discretization and SIMPLE pressure-velocity coupling. The FMG initialization scheme is employed, with coarsening of the grid to 100 cells. The one-equation Spalart-Allmaras turbulence closure model is selected to compute the eddy viscosity. The advantage of this model over two-equation turbulence models is that its robust convergence rate makes it suitable for high fidelity aerodynamic flows over complex geometries. Presented in Figure 7.54 is the accuracy of the computational flow solver compared to experimental data acquired from the NASA Langley Research Center [Ying *et al.*, 1999]. Excellent agreement is observed across the entire geometry, although the suction peaks predicted by the computational solver are marginally higher.



(a) Stream pattern



(b) Velocity contour

Figure 7.53: Contours illustrating the inviscid and viscous flow phenomena of high-lift flow

Variable screening

Global Kriging models are constructed for the aerodynamic coefficients from a stratified sample of $N_K = 100$ design points based on an LHS design. The global Kriging models are initially trained via cross-validation. Illustrated in Fig. 7.55 are the cross-validation curves for the Kriging models. The subscripts to the aerodynamic coefficients refer to the respective angle of incidence.

It is observed in Figures 7.55(a) and 7.55(b) that the constructed Kriging models for the aerodynamic coefficients at $\alpha = 8°$ are able to reproduce the training samples with sufficient confidence, recording error margin values of 3.21% and 3.83% respectively. Figure 7.55(c) demonstrates that the largest error margin occurs in predicting the maximum lift (i.e. $C_{l_{19}}$), since at larger angles of incidence the non-linear effects of flow separation and stall begin to dominate. In order to model a more correlated landscape, the sample from Figure 7.55(c) is conditioned, by eliminating all configurations which exhibit premature stall characteristics, such as unsteady flow. By conditioning the training sample,

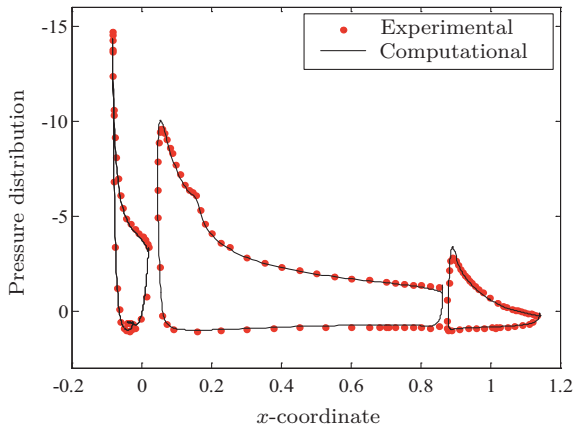
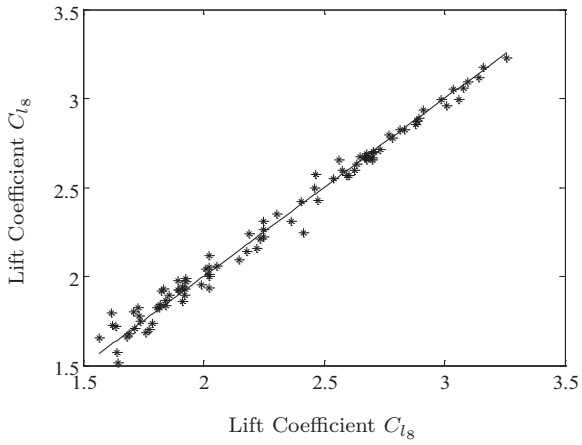


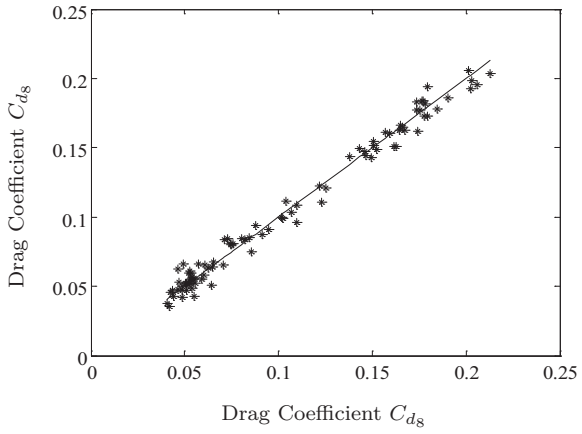
Figure 7.54: Comparison of experimental and computational surface C_p plots for the 30P/30N configuration at Re $= 9 \times 10^6$, $\alpha = 19°$

the design points are reduced to $N_K = 91$ and the error margin is reduced from 9.64% to 7.07%, which is highlighted in Figure 7.56.

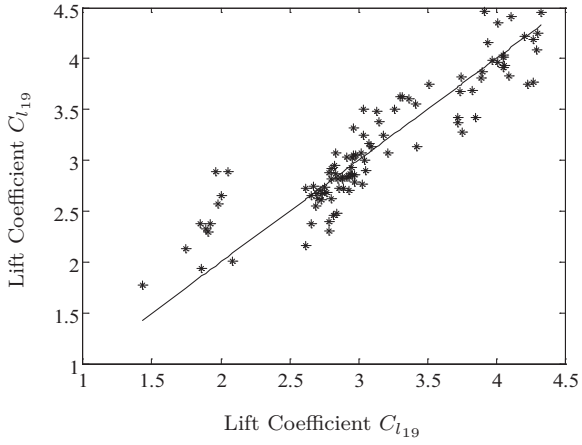
To investigate the effect of each design variable on the objective landscapes, a quantitative visualization analysis via the Morris method is conducted. Figure 7.57 features the results obtained from the design variable screening study. It is immediately observed that in the approach condition (i.e. $\alpha = 8°$) the lift generated is almost entirely dependent on the position and orientation of the flap. This is expected, since the flap component is primarily responsible for increasing the loading of the high-lift configuration. The position of the slat has a more significant effect on the drag (and most importantly Δx_S) since this may effect the development of the confluent boundary layer and thus the shear stress distribution. The slat effects become much more significant at the maximum lift condition (i.e. $\alpha = 19°$). This is due to the fact that the slat component is primarily responsible for reducing the leading edge suction peak of the trailing elements, thus delaying flow separation. The slat influence is almost completely dictated by Δx_S and $\Delta\theta_S$ suggesting that Δy_S has no elementary effect on the maximum lift generated. Similar deductions can be made by examining the variable influence on d_z shown in Figure 7.57(d). The variable influence on d_z is case-specific and entirely dependent on the reference point chosen for the proposed optimization study. Since the value of d_z is a means of ranking the success of a multiobjective solution as one single scalar, variables may be ranked by influence, which is otherwise not possible when considering a multiobjective array. Preliminary conclusions to the priority weighting of the objectives to the reference point compromise can also be made. It is important to observe that slat effects are significant in the computation of d_z, which suggests that the maximum lift condition drives the design.



(a) Training sample for C_{l_8}



(b) Training sample for C_{d_8}



(c) Training sample for $C_{l_{19}}$

Figure 7.55: CV curves for the constructed Kriging models

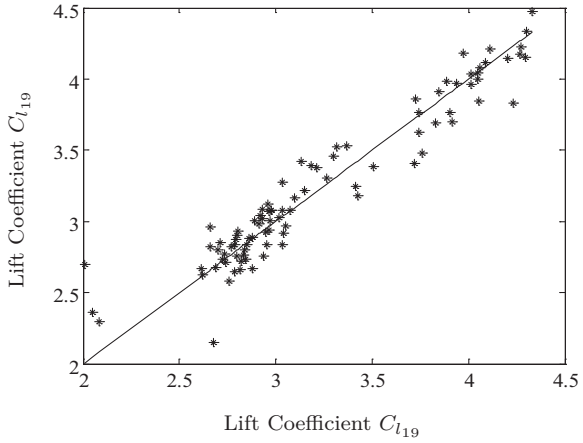


Figure 7.56: CV curve for conditioned sample for $C_{l_{19}}$

Parameter	$\mu^*(C_{l_8})$ [%]	$\mu^*(C_{d_8})$ [%]	$\mu^*(C_{l_{19}})$ [%]	$\mu^*(d_z)$ [%]	**Relative Rank**
Δx_S	<0.01	5.98	11.27	5.60	5
Δy_S	0.78	3.34	<0.01	0.34	6
$\Delta\theta_S$	2.32	0.69	19.00	9.58	4
Δx_F	30.76	25.20	27.42	31.84	1
Δy_F	33.03	19.71	13.45	23.34	3
$\Delta\theta_F$	33.12	45.08	28.84	29.28	2

Table 7.15: Results of the design variable screening study and variable ranking

Table 7.15 consolidates the results of the design variable screening study, by ranking variable importance according to its influence on d_z. It is observed that the flap variables are the most significant variables, contributing to approximately 85% of the total influence. Once again, slat effects are almost entirely dependent on Δx_S and $\Delta\theta_S$, with Δy_S contributing to less than 1% of the total influence. In an industry setting, this result could be used to reduce the dimensionality of the problem by omitting the least influential variable Δy_S, which could potentially facilitate the optimization process. In this setting the variable Δy_S is not omitted since its negligible effect is a counter-intuitive result, and may be due to insufficient sampling. Nonetheless, the true effect of Δy_S will be reconfirmed during the subsequent optimization and post-optimization processes.

Optimization results

A swarm population of $N = 100$ particles was initialized for this optimization problem. A computational budget of 250 evaluations is imposed. Based on the initial stratified sample of design points, a further 150 precise evaluations were performed over $t \approx 70$ time-steps until the computational budget was breached. As shown in Figure 7.58(a), and what

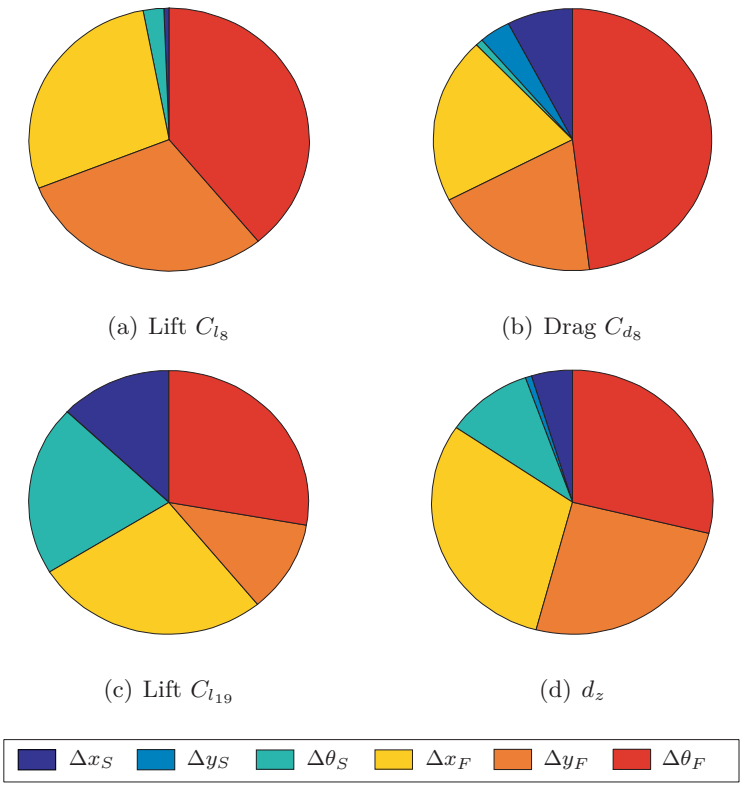
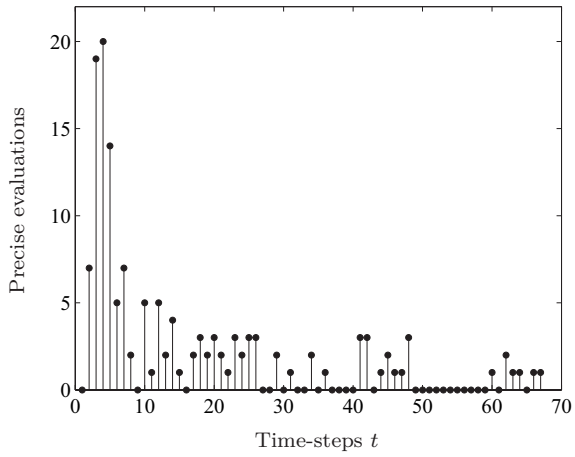


Figure 7.57: Variable influence on aerodynamic coefficients (subscripts refer to angle of incidence)

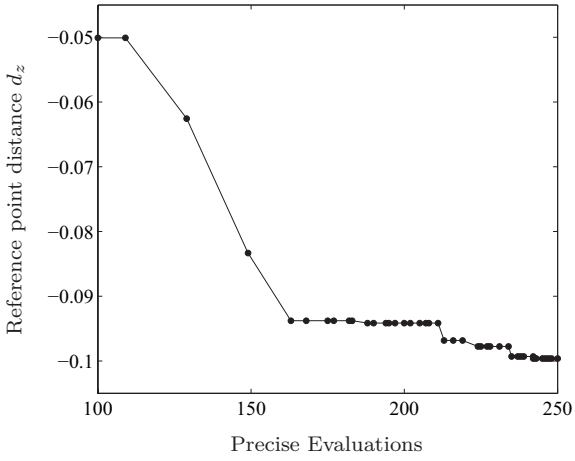
is a recurring characteristic of this algorithm, the largest number of update points are recorded during the initial explorative phase. As the preferred region is identified and the prediction error is reduced, the algorithm begins exploitation and the number of update points reduce steadily. Figure 7.58(b) features the progress of the preferred solution $\mathbf{z}'$ as the number of precise evaluations escalates. The reference point criterion is observed to filter out poorer solutions during exploration since only 50 evaluations are required to reach within 65% of the most preferred solution and an additional 50 evaluations to reach within 15%. Consistent improvement is then recorded due to exploitation of the preferred region until the search begins to converge after approximately 240 evaluations. Figure 7.59(a) features the 150 most recent solutions scheduled for precise evaluation, which once again demonstrates a preferred search trajectory. Featured in Figure 7.59(b) is the final set of 30 non-dominated solutions. The algorithm was successful in obtaining solutions which exhibit improvement over all objectives compared with the reference point.

Trade-off Visualization

Figure 7.60 features the trade-off visualization study conducted using SOMine of the identified set of non-dominated solutions shown in Figure 7.59(b). The charts are clustered



(a) History of precise updates

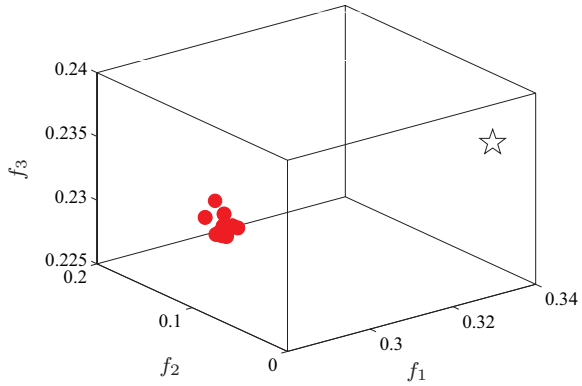


(b) Progress of most preferred solution

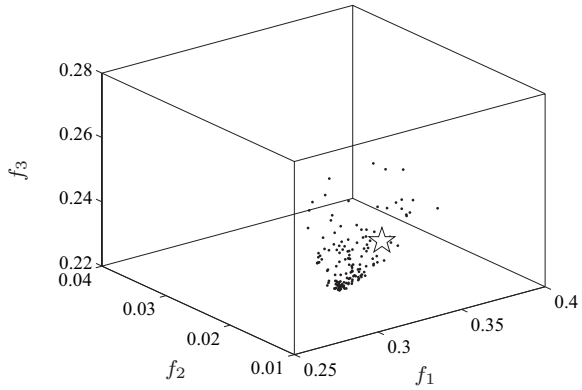
Figure 7.58: KUPMOPSO performance for high-lift optimization case-study

using eight SOM-ward clusters, organized by increasing values of d_z. It is immediately observed that the d_z chart most closely resembles the f_3 chart, indicating that this objective (i.e. maximum lift) has the highest priority. This complements the conclusion made from the variable screening analysis. In subsequent optimization studies, to identify solutions which are more inclined to the other objectives, the reference value for f_3 should be less stringent. Objectives f_1 and f_2 have a similar weighting and are within the mid-range of the identified objective boundaries.

A box-plot (see Figure 7.61) is generated of the solutions from Figure 7.60. It is observed from Figure 7.61 that the preferred region of the non-dominated front as dictated by the reference point is well defined. A reduced flap rotation angle (increased flap deflection) increases the loading by effectively increasing the camber of the airfoil configuration. The reduced slat rotation angle may effectively reduce the magnitude of flow acceleration over the slat upper surface, injecting energy into the trailing element boundary layer whilst



(a) Scatter plot of precise evaluations



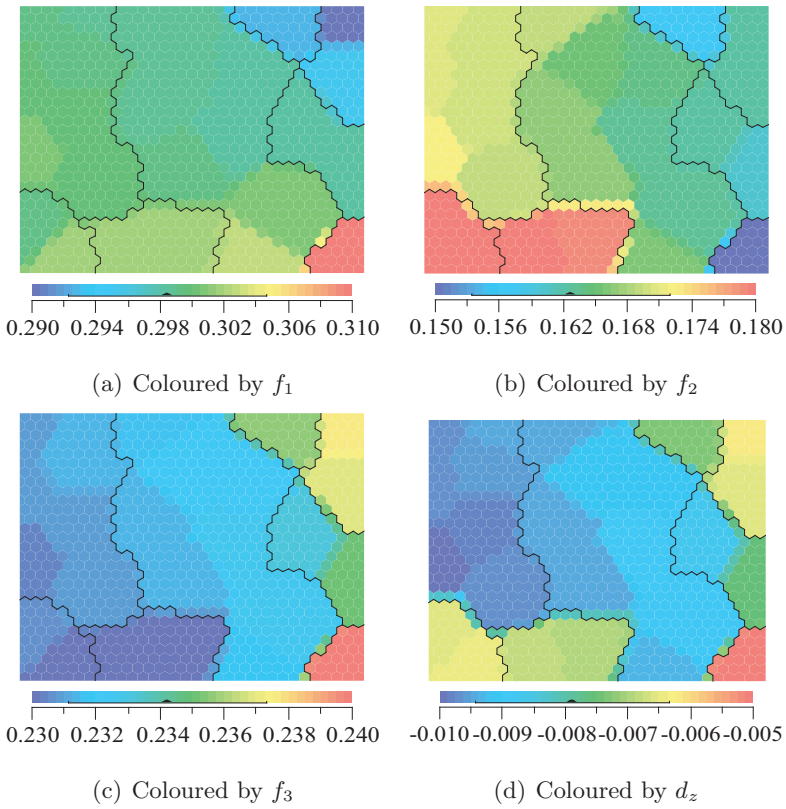
(b) Identified set of non-dominated solutions

Figure 7.59: Precise evaluations performed and the resulting non-dominated solutions

reducing the tendency of the flow to separate from the slat. The vertical translation of the flap Δy_F is at the upper bound, which could cause concern for system actuation. Both the horizontal translation of the slat and flap components are compacted such that they effectively reduce the chord, which is no doubt reflected in the increase of the lift-to-drag ratio over the reference point. The vertical translation of the slat Δy_S is the only variable shown which does not shift largely from the reference value (i.e. does not differ from the reference point 30P/30N configuration). This suggests that the variable already lies within the optimum region and thus does not have a significant impact on the dynamics of the swarm. This result coincides with the conclusion derived from the design variable screening analysis. The optimization process could confidently be re-performed with Δy_S constant at the reference value, thereby alleviating some of the computational burden.

Final designs

Despite the apparent complexity of the problem, through visualization of the SOM charts the trade-off relationships between solutions becomes quite clear. Utilizing these charts,



(a) Coloured by f_1 (b) Coloured by f_2

(c) Coloured by f_3 (d) Coloured by d_z

Figure 7.60: SOM charts to visualize optimal trade-offs between the design objectives

the designer may select the solution which best fits the target application. Figure 7.62 features the preferred configuration over the original reference configuration.

To facilitate the performance comparison between configurations, the objective functions are referred to by their aerodynamic coefficient equivalents. The values of the preferred configuration are listed in Table 7.16. Also tabulated are configurations which exhibit the most optimal value for the respective design conditions. The additional so-

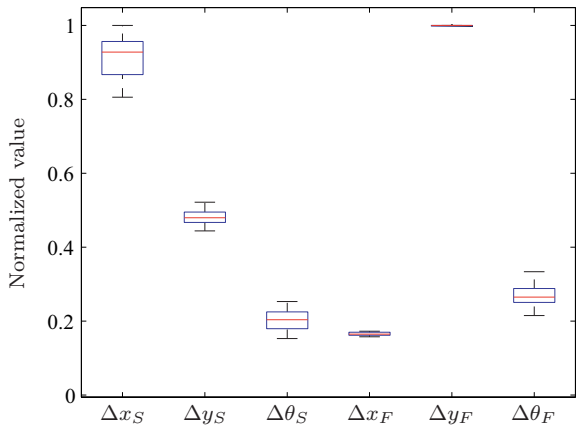


Figure 7.61: Box-plot of final non-dominated design variables

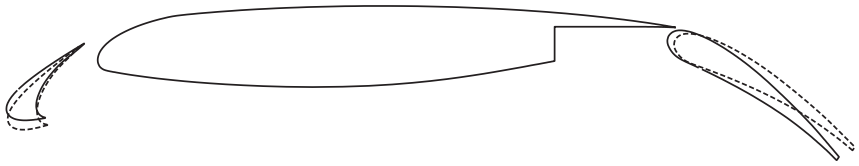


Figure 7.62: Comparison of the MDC 30P/30N configuration (- -) and the preferred solution (—)

Configuration	C_{l_8}	(C_l/C_d)	$C_{l_{19}}$
MDC 30P/30N	2.995	55.05	4.236
Preferred	3.286	58.92	4.423
min f_1	3.371	62.94	4.374
min f_2	3.194	64.84	4.335
min f_3	3.278	57.08	4.427

Table 7.16: Performance of each configuration representing the most optimal values for each design condition

lutions provide an understanding to the extent of the preferred region. The preferred solution provides an approximate 9.7%, 7.1% and 4.4% improvement over the reference 30P/30N configuration for the first, second and third design conditions respectively. The preferred solution is clearly inclined towards the $C_{l_{19}}$ condition (confirming the results of the visualization analyses) which is evident from the similarity to the min f_3 solution.

Chapter 8

Conclusion

This research was concerned with the exploration of concepts and techniques to improve the synthesis of the aerodynamic design optimization architecture. It was established that the role of the designer is of high importance in managing and facilitating optimization in a high-fidelity design environment. A method to give the designer a more active role within the optimization loop was introduced, to develop a preference-based multi-objective evolutionary algorithm capable of focusing the computational effort on identifying designs of interest. The preferences of the designer are articulated through selection of a reference point, which is a viable reflection of the preferred compromise. The use of surrogate modelling and data-mining techniques for the management of complex and high-fidelity problems was further incorporated within the framework, with favourable results. A summary of the research is hereafter presented, where the formulated research questions are revisited and the relevant findings are discussed. Avenues for further study are presented, where directions to future research which may be potentially warranted are provided.

8.1 Summary

A novel optimization framework has been developed for the application of high-fidelity aerodynamic design problems. There are three principal elements to the aerodynamic design architecture; the shape parameterization method, which governs the dimensionality of the optimization problem and is used to generate candidate shapes; the computational flow solver, which essentially dictates the fidelity of the final design and is the most computationally intensive element; and the optimization algorithm, which provides the candidate search direction and attempts to optimize the objective functions within the allocated computational budget. Evolutionary algorithms for optimization are held in high regard owing to their ability to navigate multi-modal landscapes and converge to global optima.

However such techniques are used sparingly within aerodynamic design frameworks due to the excessive number of expensive computational fluid dynamic evaluations required to achieve convergence. It is argued that improving the synthesis and efficiency of evolutionary aerodynamic design frameworks can be achieved by utilizing the domain knowledge of the designer. Control measures are therefore implemented to focus all computing effort on identifying only solutions which reflect the preferred interests of the designer.

In this research, an efficient strategy to incorporate designer preferences into a multi-objective particle swarm optimization algorithm has been presented. Particles in the swarm each represent a candidate design and co-operatively navigate the design space to identify promising regions. A reference point is stipulated by the designer, which is a reflection of the preferred compromise which can ideally be based on an existing or fictitious design. The reference point proves an effective guidance mechanism for the swarm, by selecting candidates for swarm leaders based on an inexpensive distance metric. This effectively reduces the scale of the design space, confined to exclusively focus on preferred regions. The final set of preferred non-dominated solutions are identified at a reduced computational cost over conventional search techniques. The framework is assisted by the construction of Kriging models which are progressively updated during the search. The Kriging models are updated through a novel screening criterion, which utilizes the reference point distance metric to assess the feasibility of a candidate solution for precise evaluation. Qualitative and quantitative data mining tools are applied to visualize the design space, allowing for meaningful information to be extracted which aids the pre-optimization and decision making processes. The integration of all components of the optimization framework is entirely achieved through the use of the reference point distance metric which provides a scalar measure of the success of a candidate design. Multi-objective test function suites were utilized to illustrate the operation of the algorithm and demonstrate its superiority over conventional search methods. The proposed preference-based framework was applied to a series of aerodynamic design problems of varying fidelity and complexity.

The developmental effort of the proposed framework discussed in this research is to reduce the often prohibitive computational cost of multi-objective optimization to the level of practical affordability for high-fidelity aerodynamic design problems. The formulated problems introduced focus on disparate design philosophies, utilizing inviscid potential flow theory to full transient viscous simulations. The preference-based framework is observed to perform proficiently for all case-studies within the allocated computational budget, resulting in a final set of optimized designs that are clearly reflective of the preferred

interest of the designer. The case-studies confirm the relative simplicity in assigning a reference point to identify preferred designs within the allocated computational budget.

8.2 Research Analysis

The research outcome is analyzed by first addressing the research questions formulated in Chapter 1, and subsequently highlighting the contributions and additional findings. This research has significantly added to the current body of knowledge in the area of preference-based optimization for design applications, and has offered a unique perspective to managing high-fidelity aerodynamic design problems.

8.2.1 Research Outcome

The following research questions were formulated in the introductory Chapter 1 in an attempt to consolidate the main theme of this research which is incorporating the designer within the aerodynamic design architecture for the management of high-fidelity multi-objective optimization problems. The research questions are here addressed referring to the research and study documented in previous chapters.

1. **Is there an intuitive method for identifying solutions which reflect the preferred interests of the designer?** The field of preference-based optimization was introduced in Chapter 3. With preference-based optimization it is sought to combine the searching prowess of multi-objective evolutionary algorithms with a simple and logical *a priori* method for articulating designer preferences. Providing the designer a more interactive role in the optimization architecture is a concept which became popular in the nineties. It is however only in the last decade that researchers have attempted to utilize the fast convergence properties of *a priori* methods to mitigate or alleviate the drawbacks of multi-objective evolutionary methods. In this research, a relatively new evolutionary multi-objective algorithm based on the theory of particle swarm intelligence has been proposed. The designer may provide a true reflection of their preferred interests by stipulating a reference point on the Pareto landscape, which is an intuitive method of specifying the preferred compromise and does not represent the target or *goal* solution. This reference point is used as a guidance mechanism for the swarm, such that swarm leaders are selected based on the information provided by the reference point at each time-step. This ensures that the search trajectory of the swarm remains consistent and the preferred region of the Pareto front is quickly identified by expending less computational effort.

2. **Can this method be applied to aerodynamic design, such that the domain knowledge of the designer is exploited?** The reference point utilized in this framework is a true reflection of the domain knowledge of the designer. The reference point is stipulated by the designer, which can ideally be based on an existing or fictitious design. It was demonstrated that a reference design may be located in either a feasible or infeasible area of the design space, and will invariably result in the same preferred region, provided the reference point compromise is the same. This concept was highlighted in this research, and for all design case-studies it was demonstrated how preferred solutions may be identified which improve on all performance attributes of an applicable existing design, whilst maintaining a similar level of compromise. Designs are ranked in terms of preference through an inexpensive reference point distance metric, which provides an effective guidance mechanism for the swarm, by selecting candidates for swarm leaders. A preferred region is introduced, where the extent of the region is a measure of the confidence of the designer in the reference point. If the confidence in the reference point is high, the extent of the preferred region may be reduced, such that only the most preferred designs are identified. Alternatively at the expense of increased computational effort, the extent of the preferred region may be larger, offering other interesting alternatives to the designer. Visualization techniques are also introduced to extract case-specific information, based on the preferred interests of the designer. The increase in computational efficiency is attributed to the fact that the framework utilizes the knowledge possessed by the designer. The solutions obtained are hence reflective of the preferred interests of the designer.

3. **Can it be applied to the level of practical affordability for high-fidelity design?** The benefits of evolutionary algorithms are continuously reported in the literature, especially in the context of multi-objective problems. Its unpopularity for aerodynamic design problems is due to the often excessive number of simulations required to achieve convergence. Simulations which, in order to achieve the desired levels of accuracy, are sometimes measured in hours or days for an industry-type setting. Surrogate modelling has been adopted by many researchers in aerodynamic design to alleviate the computational burden of multi-objective design. In Chapter 6, surrogate techniques were introduced and incorporated within the preference-based framework for the management of high-fidelity problems. Driving a surrogate-assisted particle swarm towards a sector of special interest on the Pareto front was shown to be an effective searching mechanism. A distinct attraction towards the preferred region is observed for all experiments, which implies the reference point

criterion is adept at filtering out solutions which deviate from the optimal search path. In this way, the full potential of the algorithm may be utilized without the risk of an excessive computational effort. Visualization tools are also applied to screen variable elementary influence and quantify their relative influence to the preferred interests of the designer. Initial design drivers can hence be easily identified. Data mining tools are also applied to facilitate a qualitative trade-off visualization study. These analyses provides an insight into the relative priority of each objective and their influence on the preferred compromise, and with the aid of the integrated surrogate model can be achieved with minimal computational effort.

8.2.2 Contributions

The major contributions extracted from this research were highlighted in Chapter 1. These contributions are briefly revisited in this section, to provide coherence to this thesis report.

- The multi-objective particle swarm optimization algorithm: provides a superior convergence rate over other multi-objective methods, and the integrated Gaussian mutation operator is ideal for the management of highly multi-modal problems.

- Integration of the reference point method: provides additional guidance to the optimizer to converge to the preferred region of the Pareto front, thereby focusing all computing effort on identifying solutions of interest. The reference point is tailored to reflect a target or existing design, and thus is used to place performance priority on specific design conditions.

- Integration of surrogate modelling: novel attempt at incorporating surrogate modelling within the preference-based (aerodynamic) optimization architecture. The search effort remains consistent, with the added advantage that less precise evaluations are performed.

- Development of a novel screening criterion: this novel criterion determines, with sufficient confidence, which candidate designs are feasible for precise evaluation. It is proficient in identifying solutions which are expected to provide improvement within the preferred region of the design space.

- The practical application of visualization tools: with the aid of the reference point metric, which measures the success of a solution as a single scalar rather than an array of objective values, the use of data mining and screening techniques to visualize a multi-objective optimization landscape is facilitated.

- Synthesis of the aerodynamic design architecture: achieved by incorporating all components of the design framework to the reference point metric which is a measure of the domain knowledge and preferred interests of the designer.

- Practical application of the algorithm: a wide range of aerodynamic design problems of varying fidelity and complexity have been considered to demonstrate the performance of the optimization framework.

8.2.3 Additional Findings and Interest

Beyond the research contributions which were documented in Chapter 1 and briefly revisited in the previous section, additional findings from this research have been identified which offer some novel or interesting concepts that are applicable to the current body of knowledge.

- Visualizing two-dimensional contour projections of the multi-dimensional design space is a fairly straightforward procedure to determine the multi-modality of a design landscape. Through analysis of the contour projections it is possible to determine whether or not a variable screening study is feasible.

- The reference point screening criterion is shown to be adept at filtering out poorly performing solutions during the search. This is evident from the scatter plots generated for the design case-studies. A distinct attraction towards the preferred region dictated by the reference point is evident, which progressively becomes localized as the search continues.

- The mutation operator, although used sparingly in this research, is demonstrated to be non-destructive. The mutation operator becomes active as the search stagnates and the swarm is no longer identifying update solutions via the conventional search techniques. It effectively manages to generate enough diversity to avoid premature convergence.

- Monitoring the progress of the most preferred design or the preferred region spread provides a crude, but reasonable measure of convergence. For most case-studies it was demonstrated that convergence was observed within the allocated computational budget by plotting the reference point distance value of the most preferred solution after each additional update evaluation. From these plots, it was possible to extract meaningful results such as the rate of convergence or the balance between exploration and exploitation.

- The initial accuracy of the constructed global Kriging models is by no means a measure of the complexity of the optimization problem. It was demonstrated that Kriging models constructed by limited samples, thereby resulting in high prediction errors, allows for more update evaluations to be allocated during the search. Considering that the Kriging models are only expected to predict solutions within the localized preferred region, it is advisable to keep the number of samples required to construct the global Kriging models to a minimum.

- Certain case-studies presented in this research were, wherever practical, complemented with the use of visualization techniques to extract case-specific information. It was demonstrated that in the majority of instances, the conclusions derived from each visualization technique complemented each other.

- Box-plots provide a suitable technique to measure the extent of a non-dominated solution spread. Through a statistical representation of the non-dominated solution variable set, it is possible to visualize the localized preferred region of the design space. While box-plots do not provide any information regarding objective comparison, they do provide a means of envisioning the geometrical profiles of the non-dominated solution set, such that conclusions can be made on the surrounding flow-field phenomena.

- During the course of optimization it may be necessary to shift the preferred region without the added computational expense of reinitializing the swarm. It was demonstrated in this research that the search direction of the swarm is able to deviate at any time-step during flight by manipulating the reference point. Despite the abrupt change in search direction, the swarm can quickly adapt to the change in preferences and converge to the updated preferred region.

- The Kriging method is utilized to construct a fairly accurate representation of a discontinuous objective landscape. This is due to the fact that it is only of interest to model the preferred region, rather than the global landscape. This favourably results in a larger number of samples focused within a localized area of the design space, thereby facilitating the model construction.

8.3 Recommendations for Future Study

This research describes an attempt at improving the synthesis of the aerodynamic design architecture via the development of a preference-based optimization framework. The

performance of this framework was highlighted through the application of a series of aerodynamic and multi-disciplinary problems of varying fidelity and complexity. While the proposed framework is complete, certain areas could be further explored or improved. This section provides a list of potential avenues for future research:

- When examining mathematical problems or problems where the preferred region of the Pareto front is known, a procedure for establishing and monitoring convergence was introduced using the Hyper-Volume metric. In the case of high-fidelity aerodynamic design problems, convergence was dictated once a computational budget had been breached. This method at this time seems the most logical choice for an industry-type setting. Convergence may be estimated by monitoring progress of the most preferred solution, or the total variance of solutions within the preferred region, but a mathematical proof of convergence is lacking which truly reflects whether the swarm has converged to the Pareto front.

- In the reported problems, it was observed that the preference-based framework was able to locate novel designs which improved on the reference point design whilst maintaining a similar level of compromise. While the framework is capable of exploring the concept of multiple reference points, its advantages or deficiencies were not examined in this research. Multiple reference points could be of practical importance in certain applications where the selection of a suitable reference point is ambiguous or the design space is ill-defined, or problems where multiple preferred solutions may be required.

- The aerodynamic problems which have been reported in this research are more inclined towards conceptual design and are not truly representative of an industrial-type multi-disciplinary problem. The proposed framework is developed for multi-objective problems and the preference-based strategy is tailored to efficiently navigate multi-dimensional landscapes. It has been demonstrated that the benefits of implementing user-preferences become more apparent as the number of objectives increase. Therefore, the most logical avenue for future research is the application of large-scale problems consisting of many objectives and many variables. Certain aspects of the framework, such as the use of the dominance criterion or the Kriging method, may not be suitable for large-scale problems. Other suitable techniques may be explored for the management of true multi-disciplinary problems.

- This framework proposed the construction of Kriging models for the management of high-fidelity problems subject to a computational budget. For simplicity the origi-

nal Kriging method was utilized, where a global training dataset was maintained to estimate the fitness at unobserved locations. However, for certain applications, it may be prudent to consider other surrogate modelling methods. For example, Artificial Neural Networks (ANN) allow for the construction of high-dimensional models involving large training datasets. Alternatively the use of a co-Kriging methodology may be beneficial for computationally intensive simulations, where lower-fidelity substitutes may be utilized to obtain a crude approximation to the high-fidelity fitness landscape. Certain surrogate methods may be specifically suited to certain objectives within a multi-objective environment. Future research in managing various surrogate models (which has already been given proposed for multi-disciplinary studies) may also be of particular significance in managing multi-fidelity problems.

- The online Kriging update criterion developed to screen predicted particles for precise evaluation utilizes the lower-bound confidence strategy. Therefore only particles which are expected to provide improvement over the current most preferred solution are scheduled for precise evaluation. While this strategy provides favourable convergence rates, the performance of other screening criteria has not been substantially provided. The expected improvement strategy could clearly provide further advantages. It was discussed in Section 5.3.2 that implementing the original expected improvement strategy for this framework is difficult due to the ambiguity associated in defining a threshold to the number of particles which may be scheduled for evaluation. Furthermore, the expected improvement criterion guides the swarm to explore areas which offer improvement in either objective rather than focusing on the preferred compromise. Future research could be directed towards developing a strategy which predicts the expected improvement in the reference point distance rather than any specific objective.

- The reference points selected for the case-studies presented in this research have generally been selected based on a previous design, or initial geometrical shape which resembles an existing design. A possible avenue for future research could involve a more thorough analysis of the influence of multiple reference point locations with a view to formulate a selection process for a suitable reference point tailored for each design case-study.

- A final avenue for future research would be to further explore and evaluate the influence of visualization tools for optimization. In this research, visualization and variable screening tools were predominately utilized to confirm the relative influence of the design variables on the optimization landscape. The conclusions derived from

these analyses could however be further utilized to facilitate the optimization process by omitting variables which have a negligible influence and assist in making informed decisions which can potentially minimize the scale of the optimization problem.

These recommendations explore only certain possible avenues for future research within the field of preference-based optimization for aerodynamic and aerospace design. It is also of the opinion of the author that visualization is a field which is relatively still in its infancy, and may offer a very interesting topic especially in the context of design. It is clear that preference-based optimization and visualization has the potential to offer many other improvements to the canonical aerodynamic design architecture, beyond those investigated in this research.

Appendix A

Mathematical Test Functions

In this appendix all test functions are described sequentially as they appear in this thesis (except those which have been described in the text). For the multi-objective ZTD and DTLZ test suites, only problems which have been considered in this thesis are reproduced. The reader is referred to the original authors for details on all remaining functions. All functions are to be minimized unless otherwise stated.

Single-objective Test Functions

The single-variable multi-modal function is given as:

$$f = (6x - 2)^2 \cdot \sin(6x - 2)^2 \quad \text{where } 0 \leq x \leq 1 \tag{A.1}$$

The convex function of *De Jong* is given as:

$$f = \sum_{i=1}^{n} x_i^2 \quad \text{where } -5.12 \leq x,\ y \leq 5.12 \tag{A.2}$$

The multi-modal function of *Rastrigin* is given as:

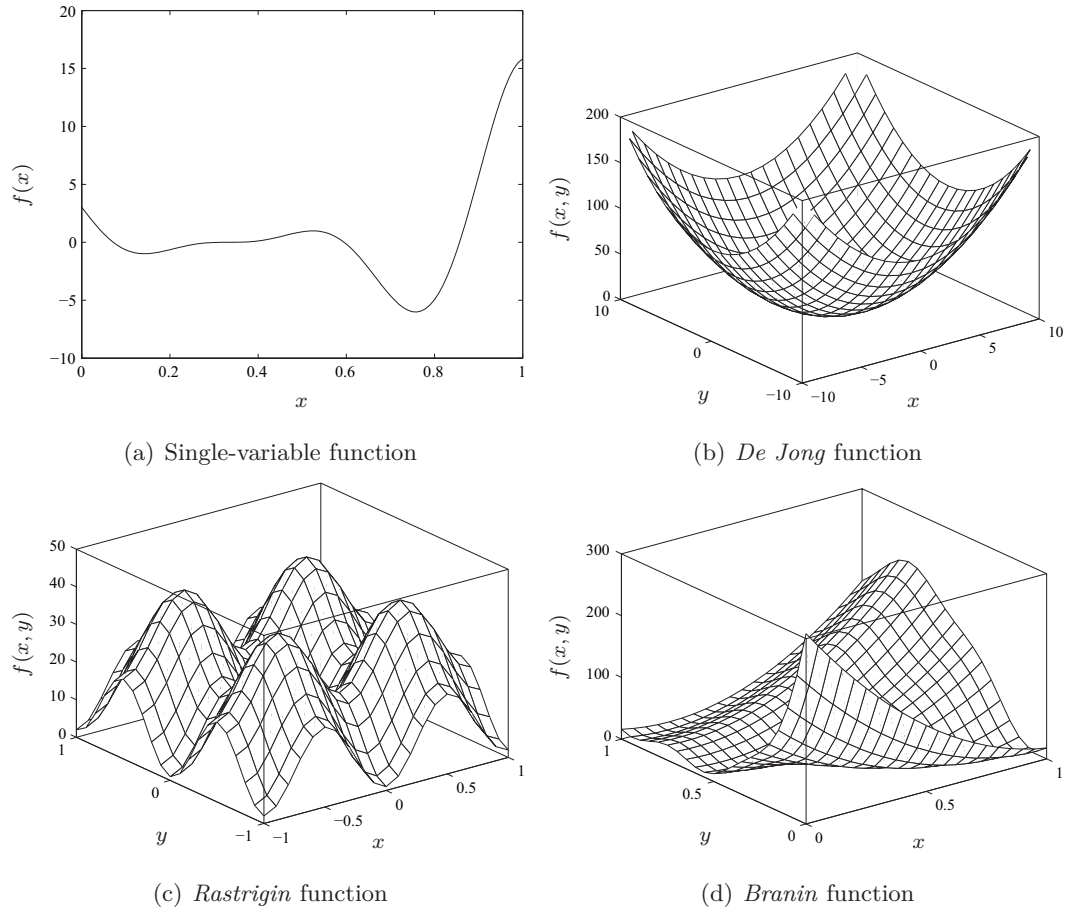
$$f = 10 \cdot n + \sum_{i=1}^{n} \left(x_i^2 - 10 \cos 2\pi x_i \right) \quad \text{where } -5.12 \leq x,\ y \leq 5.12 \tag{A.3}$$

The multi-modal function of *Branin* is given as:

$$f(x_1, x_2) = a \cdot \left(x_2 - b \cdot x_1^2 + c \cdot x_1 - d \right)^2 + e \cdot (1 - f) \cdot \cos x_1 + e$$
$$\text{where } -5 \leq x_1 \leq 10,\ 0 \leq x_2 \leq 15 \tag{A.4}$$

$$a = 1,\ b = \frac{5.1}{4\pi^2},\ c = \frac{5}{\pi},\ d = 6,\ e = 10,\ f = \frac{1}{8\pi} \tag{A.5}$$

These single-objective functions are graphically shown in Figure A.1.



(a) Single-variable function

(b) *De Jong* function

(c) *Rastrigin* function

(d) *Branin* function

Figure A.1: Single-objective test functions

The example analytical expression of the conceptual estimate of the weight of a light aircraft wing (as extracted from Forrester *et al.* [2008]) is given as:

$$W = 0.036 S_W^{0.758} W_{fw}^{0.0035} \left(\frac{A}{cos^2\Lambda}\right)^{0.6} q^{0.006} \lambda^{0.04} \left(\frac{100 t_c}{cos\Lambda}\right)^{-0.3} (N_z W_{dg})^{0.49} + S_w W_p. \quad \text{(A.6)}$$

ZTD Test Suite

The general format of the test functions is given as:

$$f_1 = x_1 \quad \text{(A.7)}$$

$$f_2 = g(x) \cdot h(x) \quad \text{(A.8)}$$

The characteristic functions of the ZTD1 test problem are:

$$g(x) = 1 + 9 \sum_{i=2}^{n} \frac{x_i}{n-1} \quad \text{(A.9)}$$

$$h(x) = 1 - \sqrt{\frac{f_1}{g(x)}} \quad \text{(A.10)}$$

where $n = 30$ and $0 \leq x_i \leq 1$. The ZTD1 test problem has a convex Pareto front, which is obtained when $g(x) = 1$.

The characteristic functions of the ZTD2 test problem are:

$$g(x) = 1 + 9 \sum_{i=2}^{n} \frac{x_i}{n-1} \tag{A.11}$$

$$h(x) = 1 - \left(\frac{f_1}{g(x)} \right)^2 \tag{A.12}$$

where $n = 30$ and $0 \leq x_i \leq 1$. The ZTD2 test problem has a concave Pareto front, which is obtained when $g(x) = 1$.

The characteristic functions of the ZTD3 test problem are:

$$g(x) = 1 + 9 \sum_{i=2}^{n} \frac{x_i}{n-1} \tag{A.13}$$

$$h(x) = 1 - \sqrt{\frac{f_1}{g(x)}} - \left(\frac{f_1}{g(x)} \right) \sin 10\pi f_1 \tag{A.14}$$

where $n = 30$ and $0 \leq x_i \leq 1$. The ZTD3 test problem has a discontinuous Pareto front, which is obtained when $g(x) = 1$.

The characteristic functions of the ZTD4 test problem are:

$$g(x) = 1 + 10(n-1) + \sum_{i=2}^{n} \left(x_i^2 - 10 \cos 4\pi x_i \right) \tag{A.15}$$

$$h(x) = 1 - \sqrt{\frac{f_1}{g(x)}} \tag{A.16}$$

where $n = 10$, $0 \leq x_1 \leq 1$ and $-5 \leq x_i \leq 5$. The ZTD4 test problem has a convex Pareto front, which is obtained when $g(x) = 1$. This problem is highly multi-modal with 21^9 local fronts.

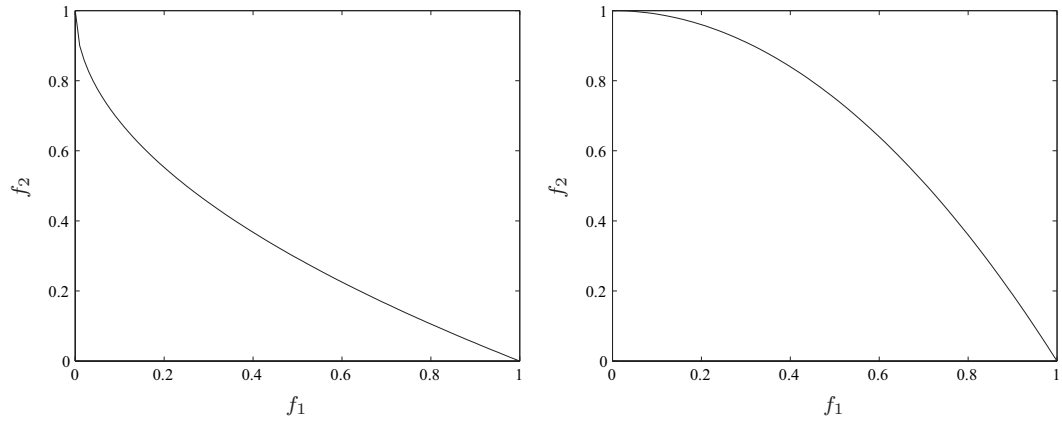
The characteristic functions of the ZTD6 test problem are:

$$f_1 = 1 - \exp -4x_1 \sin^6 6\pi x_1 \tag{A.17}$$

$$g(x) = 1 + 9 \left(\frac{(\sum_{i=2}^{n} x_i)}{n-1} \right)^{0.25} \tag{A.18}$$

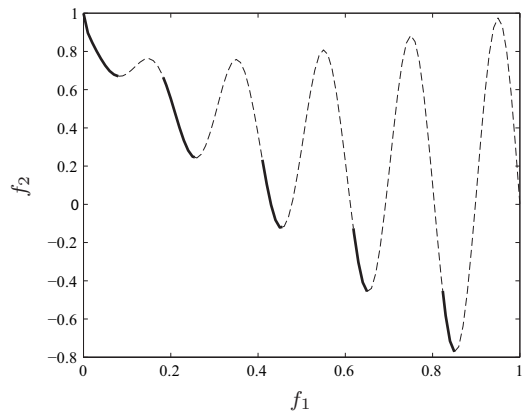
$$h(x) = 1 - \left(\frac{f_1}{g(x)} \right)^2 \tag{A.19}$$

where $n = 10$ and $0 \leq x_i \leq 1$. The ZTD6 test problem has a concave Pareto front, which is obtained when $g(x) = 1$.

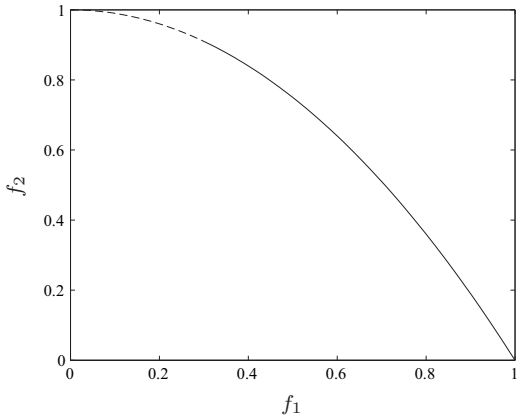


(a) ZTD1 and ZTD4

(b) ZTD2



(c) ZTD3



(d) ZTD6

Figure A.2: ZTD test problem suite

DTLZ Test Suite

The DTLZ test suite is scalable to any number of objectives and decision variables. The function g is used to generate the objectives and is defined by a subset of the decision variable vector. The subset is given as $\mathbf{x}_m = [x_m, x_{m+1}, \ldots, x_n]$ where $0 \leq x_i \leq 1$.

The characteristic functions of the DTLZ1 test problem are:

$$f_1 = \frac{1}{2} x_1 x_2 \ldots x_{m-1}(1+g) \tag{A.20}$$

$$f_2 = \frac{1}{2} x_1 x_2 \ldots (1-x_{m-1})(1+g) \tag{A.21}$$

$$\vdots$$

$$f_{m-1} = \frac{1}{2} x_1 (1-x_2)(1+g) \tag{A.22}$$

$$f_{m-1} = \frac{1}{2}(1 - x_1)(1 + g) \tag{A.23}$$

$$g = 100\left[\|\mathbf{x}_m\| + \sum_{x_i \in \mathbf{x}_m} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))\right] \tag{A.24}$$

where the number of variables $n = m + 4$ and $\|\mathbf{x}_m\| = 5$. The DTLZ1 test problem has a linear Pareto front, which is obtained when $\mathbf{x}_m = 0$. This problem is highly multi-modal with $11^5 - 1$ local fronts.

The characteristic functions of the DTLZ2 test problem are:

$$f_1 = (1 + g)\cos\left(x_1\frac{\pi}{2}\right)\cos\left(x_2\frac{\pi}{2}\right)\ldots\cos\left(x_{m-2}\frac{\pi}{2}\right)\cos\left(x_{m-1}\frac{\pi}{2}\right) \tag{A.25}$$

$$f_2 = (1 + g)\cos\left(x_1\frac{\pi}{2}\right)\cos\left(x_2\frac{\pi}{2}\right)\ldots\cos\left(x_{m-2}\frac{\pi}{2}\right)\sin\left(x_{m-1}\frac{\pi}{2}\right) \tag{A.26}$$

$$f_3 = (1 + g)\cos\left(x_1\frac{\pi}{2}\right)\cos\left(x_2\frac{\pi}{2}\right)\ldots\sin\left(x_{m-2}\frac{\pi}{2}\right) \tag{A.27}$$

$$\vdots \tag{A.28}$$

$$f_{m-1} = (1 + g)\cos\left(x_1\frac{\pi}{2}\right)\sin\left(x_2\frac{\pi}{2}\right) \tag{A.29}$$

$$f_{m-1} = (1 + g)\sin\left(x_1\frac{\pi}{2}\right) \tag{A.30}$$

$$g = \sum_{x_i \in \mathbf{x}_m} (x_i - 0.5)^2 \tag{A.31}$$

where $n = m + 9$ and $\|\mathbf{x}_m\| = 10$. The DTLZ2 test problem has a concave Pareto front, which is obtained when $\mathbf{x}_m = 0$.

The characteristic functions of the DTLZ3 test problem are:

$$f_1 = (1 + g)\cos\left(x_1\frac{\pi}{2}\right)\cos\left(x_2\frac{\pi}{2}\right)\ldots\cos\left(x_{m-2}\frac{\pi}{2}\right)\cos\left(x_{m-1}\frac{\pi}{2}\right) \tag{A.32}$$

$$f_2 = (1 + g)\cos\left(x_1\frac{\pi}{2}\right)\cos\left(x_2\frac{\pi}{2}\right)\ldots\cos\left(x_{m-2}\frac{\pi}{2}\right)\sin\left(x_{m-1}\frac{\pi}{2}\right) \tag{A.33}$$

$$f_3 = (1 + g)\cos\left(x_1\frac{\pi}{2}\right)\cos\left(x_2\frac{\pi}{2}\right)\ldots\sin\left(x_{m-2}\frac{\pi}{2}\right) \tag{A.34}$$

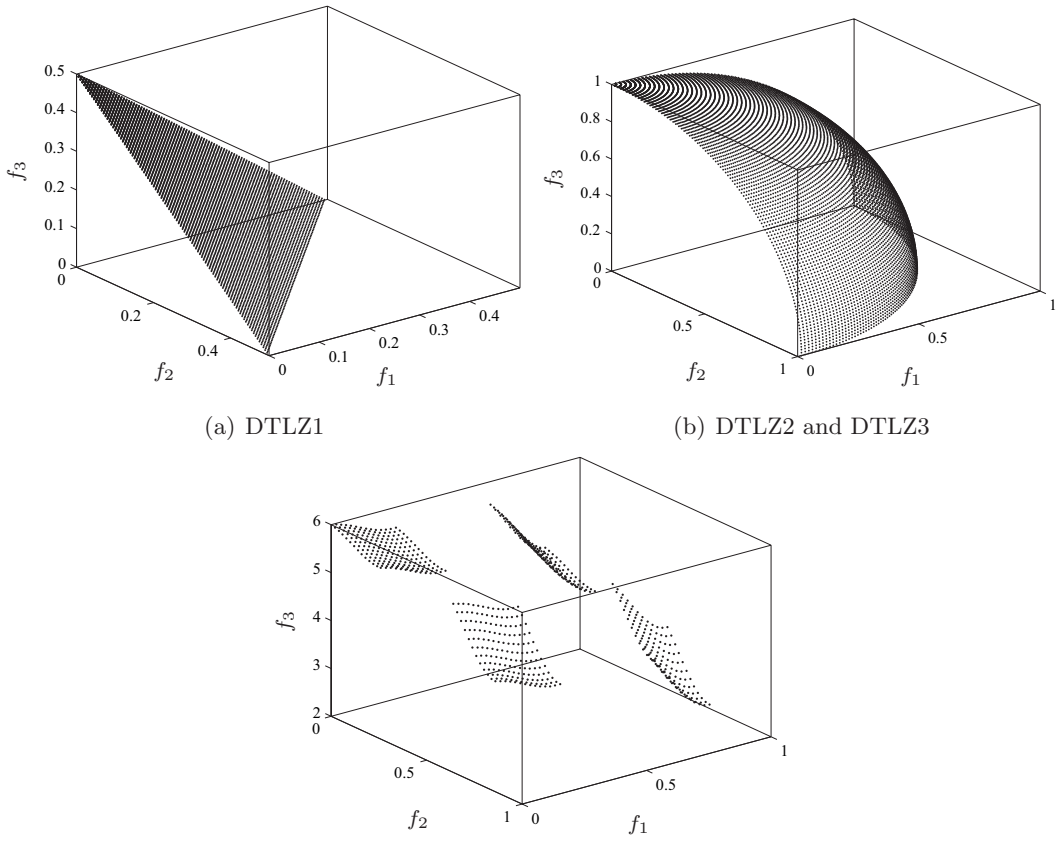
$$\vdots \tag{A.35}$$

$$f_{m-1} = (1 + g)\cos\left(x_1\frac{\pi}{2}\right)\sin\left(x_2\frac{\pi}{2}\right) \tag{A.36}$$

$$f_{m-1} = (1 + g)\sin\left(x_1\frac{\pi}{2}\right) \tag{A.37}$$

$$g = 100\left[\|\mathbf{x}_m\| + \sum_{x_i \in \mathbf{x}_m} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))\right] \tag{A.38}$$

where $n = m + 9$ and $\|\mathbf{x}_m\| = 10$. The DTLZ3 test problem has a concave Pareto front, obtained when $\mathbf{x}_m = 0$. This problem is highly multi-modal with $3^{10} - 1$ local fronts.



(a) DTLZ1

(b) DTLZ2 and DTLZ3

(c) DTLZ7

Figure A.3: DTLZ test problem suite

The characteristic functions of the DTLZ7 test problem are:

$$f_1 = x_1 \tag{A.39}$$

$$f_2 = x_2 \tag{A.40}$$

$$\vdots \tag{A.41}$$

$$f_{m-1} = x_{m-1} \tag{A.42}$$

$$f_m = (1+g) \cdot h \tag{A.43}$$

$$g = 1 + \frac{9}{\|\mathbf{x}_m\|} \sum_{x_i \in \mathbf{x}_m} x_i \tag{A.44}$$

$$h = m - \sum_{i=1}^{m-1} \left[\frac{f_i}{1+g} \left(1 + \sin\left(3\pi f_i\right)\right) \right] \tag{A.45}$$

where $n = m + 19$ and $\|\mathbf{x}_m\| = 20$. The DTLZ7 test problem has a disconnected Pareto front, with 2^{m-1} disconnected regions which are obtained when $\mathbf{x}_m = 0$.

Constrained Functions

The constrained test functions used in this thesis do not pertain to any specific test suite but are rather extracted independently from the literature.

The KITA [Kita *et al.*, 1996] test function is given as:

$$f_1 = \max \quad -x_1^2 + x_2 \tag{A.46}$$

$$f_2 = \max \quad \frac{x_1}{2} + x_2 + 1 \tag{A.47}$$

$$\text{where } 0 \leq x_1,\ x_2 \leq 7$$

The constraints for the KITA problem are:

$$g_1 = \frac{x_1}{6} + x_2 - \frac{13}{2} \leq 0 \tag{A.48}$$

$$g_2 = \frac{x_1}{2} + x_2 - \frac{15}{2} \leq 0 \tag{A.49}$$

$$g_3 = \frac{5}{x_1} + x_2 - 30 \leq 0 \tag{A.50}$$

The OSY [Osycza and Kundu, 2004] test function is given as:

$$f_1 = -\left(25 \cdot (x_1 - 2)^2 + (x_2 - 2)^2 + (x_3 - 1)^2 + (x_4 - 4)^2 + (x_5 - 1)^2\right) \tag{A.51}$$

$$f_2 = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2 \tag{A.52}$$

$$\text{where } 0 \leq x_1,\ x_2,\ x_6 \leq 10,\ 1 \leq x_3,\ x_5 \leq 5,\ 0 \leq x_4 \leq 6$$

The constraints for the OSY problem are:

$$g_1 = x_1 + x_2 - 2 \geq 0 \tag{A.53}$$

$$g_2 = 6 - x_1 - x_2 \geq 0 \tag{A.54}$$

$$g_3 = 2 - x_1 + x_2 \geq 0 \tag{A.55}$$

$$g_4 = 2 - x_1 + 3 \cdot x_2 \geq 0 \tag{A.56}$$

$$g_5 = 4 - (x_3 - 3)^2 - x_4 \geq 0 \tag{A.57}$$

$$g_6 = (x_5 - 3)^2 + x_6 - 4 \geq 0 \tag{A.58}$$

The two-bar truss design problem [Deb, 2000] is given as:

$$f_1 = x_1\sqrt{16 + x_3^2} + x_2\sqrt{1 + x_3^2} \tag{A.59}$$

$$f_2 = \max(\sigma_1,\ \sigma_2) \tag{A.60}$$

$$\sigma_1 = 20\sqrt{16 + x_3^2} \cdot \frac{1}{x_1 x_3} \tag{A.61}$$

$$\sigma_2 = 80\sqrt{1 + x_3^2} \cdot \frac{1}{x_2 x_3} \tag{A.62}$$

$$\text{where } 0 \leq x_1,\ x_2 \leq 0.01,\ 1 \leq x_3 \leq 3$$

The constraint for the two-bar truss design problem are:

$$g_1 = \max(\sigma_1,\, \sigma_2) \leq 10^5 \tag{A.63}$$

The welded beam design problem [Deb, 2000] is given as:

$$f_1 = 1.10471 \cdot h^2 \cdot l + 0.04811 \cdot t \cdot b(14 + l) \tag{A.64}$$

$$f_2 = 2.1952 \cdot \frac{1}{t^3 \cdot b} \tag{A.65}$$

$$\tau = \sqrt{\frac{(\tau')^2 + (\tau'')^2 + l \cdot \tau' \cdot \tau''}{\sqrt{0.25(l^2 + (h+t)^2)}}} \tag{A.66}$$

$$\tau' = \frac{6000}{\sqrt{2} \cdot h \cdot l} \tag{A.67}$$

$$\tau'' = \frac{6000(14 + 0.5 \cdot l)\sqrt{0.25(l^2 + (h+t)^2)}}{2\sqrt{2} \cdot h \cdot l \cdot (l^2/12 + 0.25(h+t))^2} \tag{A.68}$$

$$\sigma = \frac{504000}{t^2 \cdot b} \tag{A.69}$$

$$P_c = 64746.022(1 - 0.0282346 \cdot t) \cdot t \cdot b^3 \tag{A.70}$$

$$\text{where } 0.125 \leq h,\, b \leq 5,\; 0.1 \leq l,\, t \leq 10$$

The constraint for the two-bar truss design problem are:

$$g_1 = 13600 - \tau \geq 0 \tag{A.71}$$

$$g_2 = 30000 - \sigma \geq 0 \tag{A.72}$$

$$g_3 = b - h \geq 0 \tag{A.73}$$

$$g_4 = P_c - 6000 \geq 0 \tag{A.74}$$

Bibliography

Abbot, I. H., and von Doenhoff, A. E., 1959. *Theory of wing sections*. Dover Publications.

Anderson, D. A., Tannehill, J. C., and Pletcher, R. H., 1984. *Computational fluid mechanics and heat transfer*. Hemisphere Publishing Co. and McGraw-Hill Co.

Anderson, W. K., and Venkatakrishnan, V., 1997. Aerodynamics design optimization on unstructured grids with a continuous adjoint formulation. AIAA Paper 97-0643.

Anderson, W. K., and Bonhaus, D. L., 1999. Airfoil design on unstructured grids for turbulent flows. *AIAA Journal*. 37(2), 185–191.

Anderson, Jr., J. D., 2001. *Fundamentals of aerodynamics*. McGraw-Hill.

Angeline, P. J., 1998. Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences. In *7th International Conference on Evolutionary Programming*. San Diego, CA. 601–610.

Arora, J. S., 2004. *Introduction to optimum design*. Elsevier Academic Press.

Blake, W. K., 1975. A statistical description of pressure and velocity fields at the trailing edge of a flat strut. *David Taylor Naval Ship R and D Center* Report 4241.

Blake, W. K., 1986. *Mechanics of flow-induced sound and vibration, Vol. I and II*. London: Academic Press.

Braibant, V., and Fleury, C., 1984. Shape optimal design using B-splines. *Computer Methods in Applied Mechanics and Engineering*. 44(3), 247–267.

Bratton, D., and Kennedy, J., 2007. Defining a standard for particle swarm optimization. In *Proceedings of the IEEE Swarm Intelligence Symposium*. Honolulu, HI. 120–127.

Bogue, D., and Crist, N., 2006. CST transonic optimization using Tranair++. *46th AIAA Aerospace Sciences Meeting and Exhibit*. Reno, Nevada.

Buckley, H. P., Zhou, B. Y., and Zingg, D. W., 2010. Airfoil optimization using practical aerodynamic design requirements. *Journal of Aircraft*. 47(5), 1707–1719.

Castonguay, P., and Nadarajah, S., 2007. Effect of shape parameterization on aerodynamic shape optimization. *45th AIAA Aerospace Science Meeting and Exhibit*. Reno, Nevada.

Carrese, R., and Winarto, H., 2011. An efficient strategy to incorporate designer-preferences in automated airfoil design. In *Proceedings of the 14th Australian International Aerospace Congress*. Melbourne, Australia.

Carrese, R., Winarto, H., Watmuff, J., and Wickramasinghe, U. K., 2011. Benefits of incorporating designer preference within a multi-objective airfoil design framework. *Journal of Aircraft*. 48(3), 832–844.

Carrese, R., Sóbester, A., Winarto, H., and Li, X., 2011. Swarm heuristic for identifying preferred solutions in surrogate-based multi-objective engineering design. *AIAA Journal*. 49(7), 1437–1449.

Carrese, R., Winarto, H., Li, X., Sóbester, A., and Ebenezer, S., 2012. A comprehensive preference-based optimization framework with application to high-lift aerodynamic design. *Engineering Optimization*. 1–19.

Chiba, K., Makino, Y., and Takatoya, T., 2008. Evolutionary-based multidisciplinary design exploration for the silent supersonic technology demonstrator wing. *Journal of Aircraft*. 45(5), 1481–1494.

Chung, H., and Alonso, J. J., 2004. Multiobjective optimization using approximation-based genetic algorithms. In *Proceedings of the 10th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*. Albany, NY. 2004–4325.

Clerc, M., and Kennedy, J., 2002. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*. 6(1), 58–73.

Coello Coello, C. A., 2000. Handling preferences in evolutionary multiobjective optimization: A survey. In *Congress on Evolutionary Computation*. San Diego, CA. 30–37.

Coello Coello, C. A., and Lechuga, M. S., 2002. MOPSO: A proposal for multiple objective particle swarm optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation*. 2, 1051–1056.

Coello Coello, C. A., 2007. *Evolutionary algorithms for solving multi-objective problems*. Springer.

Crossley, W. A., 1996. Genetic algorithm approaches for multiobjective design of rotor systems. In *Proceedings of the 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*. Bellevue, WA. 384–394.

Curle, N., 1955. The influence of solid boundary upon aerodynamic sound. In *Proceedings of the Royal Society London A*. 231, 505–514.

D'Angelo, S., and Minisci, E.A., 2005. Multi-objective evolutionary optimization of subsonic airfoils by Kriging approximation and evolution control. In *IEEE Congress on Evolutionary Computation*. IEEE Computer Society Press. 1262–1267.

Das, I., and Dennis, J. E., 1997. A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Structural Optimization*. 14, 63–69.

Deb, K., 1999. Solving goal programming problems using multi-objective genetic algorithms. In *IEEE Congress on Evolutionary Computation*. IEEE Computer Society Press. 77–84.

Deb, K., 2000. An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*. 186(2–4), 311–338.

Deb, K., 1999. *Multi-objective optimization using evolutionary algorithms*. John Wiley and Sons, Ltd.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*. 6(2), 182–197.

Deb, K., Thiele, L., Laumanns, M., and Zitzler, E., 2005. Scalable test problems for evolutionary multi-objective optimization. In Evolutionary Multiobjective Optimization (EMO): Theoretical Advances and Applications. Springer. 105–145.

Deb, K., and Sundar, J., 2006. Reference point based multi-objective optimization using evolutionary algorithms. In *Genetic and Evolutionary Computation Conference*. ACM Press. 635–642.

Deb, K., and Kumar, A., 2007. Interactive evolutionary multi-objective optimization and decision-making using reference direction method. In *Genetic and Evolutionary Computation Conference*. ACM Press. 781–788.

Deb, K., and Kumar, A., 2007. Light beam search based multi-objective optimization using evolutionary algorithms. In *Genetic and Evolutionary Computation Conference*. ACM Press. 2125–2132.

De Boor, C., 1994. *A practical guide to splines*. Springer.

Drela, M., and Giles, M. B., 1987. Viscous-inviscid analysis of transonic and low Reynolds number airfoils. *AIAA Journal*. 25(10), 1347–1355.

Drela, M., 1998. Pros and cons of airfoil optimization. In *Frontiers of computational fluid dynamics*. World Scientific.

Eberhart, R. C., and Shi, Y., 2000. Comparing inertia weights and constriction factors in particle swarm optimization. In *Proceedings of the Congress on Evolutionary Computation*. La Jolla, CA. 84–88.

Ehrgott, M., and Gandibleux, X., 2002. Multiple criteria optimization: state of the art annotated bibliographic surveys. *International Series in Operations Research and Management Science*. 52. Kluwer Academic Publishers.

Elliott, J., and Peraire, J., 1996. Aerodynamic design using unstructured meshes. AIAA Paper. 96–1941.

Elliott, J., and Peraire, J., 1997. Practical 3D aerodynamic design and optimization using unstructured meshes. *AIAA Journal*. 35(9), 1479–1485.

Epstein, B., Jameson, A., Peigin, S., Roman, D., Harrison, N., and Vassberg, J., 2009. Comparative study of three-dimensional wing drag minimization by different optimization techniques. *Journal of Aircraft*. 46(2), 526–541.

Emmerich, M., Beume, N., and Naujoks, B., 2005. An EMO algorithm using the hypervolume measure as selection criterion. *in Evolutionary Multi-Criterion Optimization, series on Lecture Notes in Computer Science*. Springer. 3410, 62–76.

Emmerich, M., Giannakoglou, K. C., and Naujoks, B., 2006. Single- and multi-objective evolutionary optimization assisted by Gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation*. 10(4), 421–438.

Engelbrecht, A. P., 2005. *Fundamentals of computational swarm intelligence*. John Wiley and Sons, Ltd.

Fausett, L., 1994. *Fundamentals of neural networks: architectures, algorithms, and applications*. Prentice Hall.

Fieldsend, J. E., and Singh, S., 2002. A multi-objective algorithm based upon particle swarm optimisation, an efficient data structure and turbulence. In *Proceedings of the U.K Workshop on Computational Intelligence*. 37–44.

Fieldsend, J. E., 2004. Multi-objective particle swarm optimization methods. Technical Report. Department of Computer Science, University of Exeter.

Fonseca, C. M., and Fleming, P. J., 1993. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *Proceedings of the Fifth International Conference on Genetic Algorithms*. Morgan Kauffman Publishers. 416–423.

Fonseca, C. M., and Fleming, P. J., 1995. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*. 3, 1–16.

Fonseca, C. M., and Fleming, P. J., 1998. Multiobjective optimization and multiple constraint handling with evolutionary algorithms-Part I: A unified formulation. *IEEE Trans on Sys Man and Cybernetics-Part A*. 28(1), 26–37.

Forrester, A., Sóbester, A., and Keane, A., 2008. *Engineering design via surrogate modelling: A practical guide*. John Wiley and Sons, Ltd.

Gano, S. E., Renaud, J. E., Martin, J. D., and Simpson, T. W., 2006. Update strategies for Kriging models used in variable fidelity optimization. *Structural and Multidisciplinary Optimization*. 32(4), 287–298.

Garner, L., Meredith, P.T., and Stoner, R.C., 1991. Areas for CFD development as illustrated by transport aircraft applications. AIAA Paper 91-1527.

Giles, M. B., and Pierce, N. A., 2000. An introduction to the adjoint approach to design. *Flow, Turbulence and Combustion*. 65(3–4), 393–415.

Goldberg, D. E., and Richardson, J. J., 1987. Genetic algorithms with sharing for multimodal function optimization. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*. L. Erlbaum Associates Inc. 41–49.

Goldberg, D. E., 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley.

Greenman, R. M., and Roth, K. R., 1999. High-lift optimization design using neural networks on a multi-element airfoil. *Journal of Fluids Engineering*. 121(2), 434–440.

Grosan, C., Abraham, A., and Ishibuchi, H., (eds) 2007. *Hybrid evolutionary algorithms*. Springer-Verlag Berlin/Heidelberg.

Hedayat, A. S., Sloane, N. J. A., and Stufken, J., 1999. Orthogonal arrays: Theory and applications. Springer.

Hicks, R. M., and Henne, P. A., 1978. Wing design by numerical optimization. *Journal of Aircraft*. 15(7), 407–412.

Holst, T. L., and Pulliam, T. H., 2001. Aerodynamic shape optimization using a real-number-encoded genetic algorithm. AIAA Paper No. 2001-2473.

Howe, M.S., 1998. *Acoustics of fluid structure interaction*. Cambridge University Press.

Huyse, L., and Michael, L. R., 2001. Aerodynamic shape optimization of two-dimensional airfoils under uncertain operating conditions. *Institute for Computer Applications in Science and Engineering*. Technical Report.

Jahangirian, A., and Shahrokhi, A., 2009. Inverse design of transonic airfoils using genetic algorithms and a new parametric shape model. *Inverse Problems in Science and Engineering*. 17(5), 681–699.

Jameson, A., 1974. Iterative solution of transonic flows over airfoils and wings, including flows at Mach one. *Communications on Pure and Applied Mathematics*. 27(3), 283–309.

Jameson, A., 1988. Aerodynamic design via control theory. *Journal of Scientific Computing*. 3(3), 233–260.

Jameson, A., Martinelli, L., Pierce, N. A., 1998. Optimum aerodynamic design using the Navier-Stokes equations. *Theoretical and Computational Fluid Dynamics*. 10(1–4), 213–237.

Jameson, A., 2003. Aerodynamic shape optimization using the adjoint method. *VKI Lecture Series on Aerodynamic Drag Prediction and Reduction*. von Karman Institute of Fluid Dynamics. 3–7.

Jeong, S., Murayama, M., and Yamamoto, K., 2005. Efficient Optimization Design Method Using Kriging Model. *Journal of Aircraft*. 42(2), 413–420.

Jeong, S., Chiba, K., and Obayashi, S., 2005. Data mining for aerodynamic design space. *Journal of Aerospace Computing, Information, and Communication*. 2, 452–469.

Jin, Y., Olhofer, M., and Sendhoff, B., 2001. Dynamic weighted aggregation for evolutionary multiobjective optimization: Why does it work and how?. In *Proceedings of Genetic and Evolutionary Conference*. San Francisco, 1042–1049.

Jin, Y., Olhofer, M., and Sendhoff, B., 2002. A framework for evolutionary optimization with approximate fitness functions. *IEEE Transactions on Evolutionary Computation*. 6(5), 481–494.

Jin, Y., 2005. A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing*. 9(1), 3–12.

Jin, Y., and Sendhoff, B., 2009. A systems approach to evolutionary multi-objective structural optimization and beyond. *IEEE Computational Intelligence Magazine*. 4(3), 62–76.

Johnson, M. E., Moore, L. M., and Ylvisakar, D., 1990. Minimax and maximin distance designs. *Journal of Statistical Planning and Inference*. 26, 131–148.

Jones, B. R., Crossley, W. A., and Lyrintzis, A. S., 1998. Aerodynamic and aeroacoustic optimization of airfoils via a parallel genetic algorithm. In *Proceedings of the 7th AIAI/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*. 1–11.

Jones, D. R., 1998. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*. 13, 455–492.

Jones, D. R., 2001. A taxomony of global optimization methods based on response surfaces. *Journal of Global Optimization*. 21, 345–383.

Kanazaki, M., Tanaka, K., Jeong, S., Yamamoto, K., 2006. Multi-objective aerodynamic optimization of elements' setting for high-lift airfoil using Kriging model. *44th AIAA Aerospace Sciences Meeting and Exhibit*. Reno, NV.

Keane, A. J., and Nair, P. B., 2005. *Computational approaches for aerospace design: The pursuit of excellence*. John Wiley and Sons, Ltd.

Kennedy, J., and Eberhart, R., 1995. Particle swarm optimization. In *Proceedings of IEEE International Conference on Neural Networks*. Piscataway, NJ. 1942–1948.

Kennedy, J., Eberhart, R. C., and Shi, Y., 2001. *Swarm intelligence*. Morgan Kaufmann Publishers.

Kennedy, J., and Mendes, R., 2002. Population structure and particle swarm performance. In *Proceedings of the Congress on Evolutionary Computation*. IEEE Press. 1671–1676.

Khurana, M., Sinha, A., and Winarto, H., 2008. Multi-mission re-configurable UAV - Airfoil optimization through swarm approach and low fidelity solver. In *Proceedings of the 23rd International Conference on Unmanned Air Vehicle Systems*. Bristol, UK.

Khurana, M., Winarto, H., and Sinha, A., 2008. Application of swarm approach and artificial neural networks for airfoil shape optimization. In *Proceedings of the 12th AIAA/ISSMO Multidisciplinary Analysis and Optimisation*. Victoria, Canada.

Khurana, M., and Winarto, H., 2010. Development and validation of an efficient direct numerical optimisation approach for airfoil shape design. *The Aeronautical Journal*. 114(1160), 611–628.

Kim, S., Alonso, J.J., and Jameson, A., 2004. Multi-element high-lift configuration design optimization using viscous continuous adjoint method. *Journal of Aircraft*. 41(5), 1082–1097.

Kita, H., Yabumoto, Y., Mori, N., and Nishikawa, Y., 1996. Multiobjective optimization by means of the thermodynamical genetic algorithm. In *PPSN IV Proceedings of the 4th International Conference on Parallel Problem Solving from Nature*. Springer-Verlag. 504–512.

Kirkpatric, S., Gelatt, C. D., and Vecchi, M. P., 1983. Optimization by simulated annealing. *Science*. 220(4598), 671–680.

Knowles, J., and Corne, D., 2000. Approximating the non-dominated front using the Pareto archived evolution strategy. *Evolutionary Computation*. 8(2), 149–172.

Knowles, J., and Corne, D., 2002. On metrics for comparing non-dominated sets. In *Congress on Evolutionary Computation*. IEEE Computer Society Press. 711–716.

Kohonen, T., 1995. *Self-organizing maps*. Springer-Verlag.

Kulfan, B. M., and Bussoletti, J. E., 2006. Fundamental parametric geometry representations for aircraft component shapes. AIAA Paper 2006-6948.

Kulfan, B. M., 2008. Universal parametric geometry representation method. *Journal of Aircraft*. 45(1), 142–158.

Labrujère, T. E., Sloof, J. W., 1993. Computational methods for the aerodynamic design of aircraft components. *Annual Review of Fluid Mechanics*. 25, 183–214.

Leary, S. J., Bhaskar, A., and Keane, A. J., 2002. Screening and approximation methods for efficient structural optimization. In *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*. Atlanta, Georgia.

Lèpine, J., Guibault, F., Trépanier, J. Y., and Pépin, F., 2001. Optimized nonuniform rational B-spline geometrical representation for aerodynamic design of wings. *AIAA Journal*. 39(11), 2033–2041.

Lee, K., Park, G., and Joo, W., 2005. A global robust optimization using the Kriging based approximation model. *JSME International Journal*. 49(3), 779–788.

Li, W., Huyse, L., and Padula, S., 2002. Robust airfoil optimization to achieve drag reduction over a range of Mach numbers. *Structural and Multidisciplinary Optimization*. 24, 38–50.

Li, X., 2003. A non-dominated sorting particle swarm optimizer for multiobjective optimization. In *Genetic and Evolutionary Computation Conference*, Chicago, IL. 37–48.

Li, X., 2004. Better spread and convergence: Particle swarm multiobjective optimization using the maximin fitness function. In *Genetic and Evolutionary Computation Conference*, Seattle, WA. 117–128.

Li, M., Li, G., Azarm, S., 2008. A Kriging metamodel assisted multi-objective genetic algorithm for design optimization. *Journal of Mechanical Design*. 130(3).

Lighthill, M. J., 1945. A new method of two dimensional aerodynamic design. *The Aeronautical Research Council*. RAND Technical Report M2112.

Lighthill, M. J., 1952. On sound generated aerodynamically, I: General Theory. In *Proceedings of the Royal Society London A*. 211, 564Ű-587.

Macabe, A., 1967. Design of a supersonic nozzle. *Ministry of Aviation: Aeronautical Research Council*. Report No. 3440.

Marco, N., Desideri, J. A., and Lanteri, S., 1999. Multiobjective optimization in CFD by genetic algorithms. *Institut National de Recherche en Informatique et en Automatique*. Report No. 3686.

Marler, R. T., and Arora, J. S., 2004. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*. 26, 369–395.

Marsden, A., Wang, M., Dennis, J. E. Jr., and Moin, P., 2004. Optimal aeroacoustic shape design using the surrogate management framework. *Optimization and Engineering*. 5, 235–262.

Mckay, M. D., Beckman, R. J., and Conover, W. J., 1979. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics. 21(2), 239–245.

Matheron, G., 1963. Principles of geostatistics. *Economic Geology*. 58, 1246–1266.

Messac, A., 1996. Physical programming: effective optimization for computational design. *AIAA Journal*. 34(1), 149–158.

Messac, A., Sukam, C. P., and Melachrinoudis, E., Aggregate objective functions and Pareto frontiers: required relationships and practical implications. *Optimization and Engineering*. 1, 171–188.

Messac, A., Sundararaj, G. J., Tappeta, R. V., and Renaud, J. E., Ability of objective functions to generate points on nonconvex Pareto frontiers. *AIAA Journal*. 38(6), 1084–1091.

Messac, A., Ismail-Yahaya, A., and Mattson, C. A., 2003. The normalized normal constraint method for generating the Pareto frontier. *Structural and Multidisciplinary Optimization*. 25, 86–98.

Mitchell, M. T., and Morris, M. D., 1992. Bayesian design and analysis of computer experiments: Two examples. *Statistica Sinica*. 2, 359–379.

Morris, M. D., 1991. Factorial sampling plans for preliminary computational experiments. *Technometrics*. 33(2), 161–174.

Morris, M. D., and Mitchell, T. J., 1995. Bayesian design and analysis of computer experiments: Two examples. *Journal of Statistical Planning and Inference*. 43, 381–402.

Mostaghim, S., and Teich, J., 2003. The role of ϵ-dominance in multi-objective particle swarm optimization methods. In *Proceedings of the IEEE Congress on Evolutionary Computation*. 1764–1771.

Montgomery, D. C., 1997. Design and analysis of experiments. John Wiley and Sons, Ltd.

Mousavi A., Castonguay P., and Nadarajah S., 2007. Survey of shape parameterization techniques and its effect on three-dimensional aerodynamic shape optimization. *18th AIAA Computational Fluid Dynamics Conference*. Miami, Florida.

Nemec, M., and Zingg, D. W., 2002. Newton-Krylov algorithm for aerodynamic design Using the Navier-Stokes equations. *AIAA Journal*. 40(6), 1146–1154.

Nemec, M., Zingg, D. W., and Pulliam, T. H., 2004. Multipoint and multi-objective aerodynamic shape optimization. *AIAA Journal*. 42(6), 1057–1065.

Niu, M. C. Y., 2002. *Airframe structural design*. Hong Kong Conmilit Press Ltd.

Norris, S. R., and Crossley, W. A., 1998. Pareto-optimal controller gains generated by a genetic algorithm. *36th AIAA Aerospace Sciences Meeting and Exhibit*. Reno, NV.

Obayashi, S., Yamaguchi, Y., and Nakamura, T., 1997. Multiobjective genetic algorithm for multidisciplinary design of transonic wing planform. *Journal of Aircraft*. 34(5), 690–693.

Obayashi, S., Tsukahara, T., and Nakamura, T., 2000a. Multiobjective genetic algorithm applied to aerodynamic design of cascade airfoils. *IEEE Transactions on Industrial Electronics*. 47(1), 211–216.

Obayashi, S., Sasaki, D., Takeguchi, Y. and Hirose, N., 2000b. Multiobjective evolutionary computation for supersonic wing-shape optimization. *IEEE Transactions on Evolutionary Computation*. 4(2), 182–187.

Ong, Y. S., Nair, P. B., and Keane, A. J., 2003. Evolutionary optimization of computationally expensive problems via surrogate modeling. *AIAA Journal*. 41(4), 687–696.

Ong, Y. S., Nair, P. B., Keane, A. J., and Zhou, Z. Z., 2004. Surrogate-assisted evolutionary optimization frameworks for high-fidelity engineering design problems. *Knowledge Incorporation in Evolutionary Computation*. Springer Verlag. 307–331.

Osycza, A., and Kundu, S., 1995. A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm. Structural Optimization. 10, 94–99.

Oyama, A., Obayashi, S., and Nakamura3, T., 2001. Real-coded adaptive range genetic algorithm applied to transonic wing optimization. *Applied Soft Computing*. 1(3), 179–187.

Painchaud-Ouellet, S., Tribes, C., Trépanier, J., and Pelletier, D., 2006. Airfoil shape optimization using a nonuniform rational B-splines parameterization under thickness constraint. *AIAA Journal*. 44(10), 2170–2178.

Pareto, V., 1927. *Manual d'économie politique*. Marcel Giard.

Pasquale, D., Harinck, J., Guardone, A., and Rebay, S., 2010. Geometry optimization for quasi-uniform flows from supersonic nozzles. In *European Conference on Computational Fluid Dynamics*. Lisbon, Portugal.

Praveen, C., and Duvigneau, R., 2009. Low cost PSO using metamodels and inexact pre-evaluation: Application to aerodynamic shape design. *Computer Methods in Applied Mechanics and Engineering*. 198, 1087–1096.

Piegel, L. A., and Tiller, W., 1996. *The NURBS book: Monographs in visual communication*. Springer.

Polhamus, E. C., Geller, E. W., and Grunwald, K. J., 1959. Pressure and force characteristics of noncircular cylinders as affected by Reynolds number with a method included for determining the potential flow about arbitrary shapes. *NASA Technical Report R-46*.

Quagliarella, D., and Della Cioppa, A., 1994. Genetic algorithms applied to the aerodynamic design of transonic airfoils. In the *12th AIAA Applied Aerodynamics Conference*. Colorado Springs, CO. 686–693.

Quagliarella, D., and Vicini, A., 2001. Viscous single and multicomponent airfoil design with genetic algorithms. *Finite Elements in Analysis and Design*. 37(5), 365–380.

Rachmawati, L., and Srinivasan, D., 2006. Preference incorporation in multi-objective evolutionary algorithms: A survey. *IEEE Congress on Evolutionary Computation*. Vancouver, Canada. 962–968.

Raquel, C. R., and Naval, P. C., 2005. An effective use of crowding distance in multiobjective particle swarm optimization. In *Proceedings of Genetic and Evolutionary Computation*. ACM Press. 257–264.

Ray, T., Tai, K., Seow, K. C., 2001. Multiobjective design optimization by an evolutionary algorithm. *Engineering Optimization*. 33(3), 399–424.

Ray, T., and Liew, K. M., 2002. A Swarm metaphor for multiobjective optimization. *Engineering Optimization*. 34(2), 141–153.

Ray, T., and Tsai, H. M., 2004. Swarm algorithm for single and multiobjective airfoil design optimization. *AIAA Journal*. 42(2), 366–373.

Reuther, J., and Jameson, A., 1995. Supersonic wing and wing-body shape optimization using an adjoint formulation. RIACS TR-95-14.

Reyez-Sierra, M., and Coello Coello, C. A., 2006. Multi-objective particle swarm optimizers: A survey of the state-of-the-art. *International Journal of Computational Intelligence Research*. 2(3), 287–308.

Reyez-Sierra, M., and Coello Coello, C. A., 2005. Improving PSO-based multi-objective optimization using crowding, mutation and ϵ-dominance. *Lecture Notes in Computer Science*. ISSU 3410. 505–519.

Rogalsky, T., Kocabiyik, S., and Derksen, R. W., 2000. Differential evolution in aerodynamic optimization. *Canadian Aeronautics and Space Journal*. 46(4), 183–190.

Rogers, S.E., Menter, F.R., Mansour, N.N., and Durbin, P.A., 1994. A comparison of turbulence models in computing multi-element airfoil flows. AIAA Paper 94-0291.

Rumpfkeil, M. P., and Zingg, D. W., 2008. Unsteady optimization using a discrete adjoint approach applied to aeroacoustic shape design. In *46th AIAA Aerospace Sciences Meeting*. Reno, Nevada.

Rumpfkeil, M. P., and Zingg, D. W., 2010. The optimal control of unsteady flows with a discrete adjoint method. *Optimization and Engineering*. 11(1), 5–22.

Rumsey, C.L., Gatski, T.B., Ying, S.X., Bertelrud, A., 1998. Prediction of high-lift flows using turbulence closure models. *AIAA Journal*. 36 (5), 765–774.

Rumsey, C.L., and Ying, S.X., 2002. Prediction of high lift: review of present CFD capability. *Progress in Aerospace Sciences*, 38 (2), 145–180.

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H., 1989. Design and analysis of computer experiments. *Statistical Science*. 4(4), 409–423.

Santer, T. J., Williams, B., and Notz, W., 2003. *The design and analysis of computer experiments*. Springer-Verlag.

Schaffer, J. D., 1989. Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the 1st International Conference on Genetic Algorithms*. Hillsdale, NJ.

Saltelli, A., Tarantola, S., Campolongo, F., and Ratto, M., 2001. *Sensitivity analysis in practice*. John Wiley and Sons, Ltd.

Santers, T. J., Williams, N. J., Notz, W. I., 2003. *The design and analysis of computer experiments*. Springer-Verlag.

Samareh, J. A., 2001. Survey of shape parameterization techniques for high-fidelity multidisciplinary shape optimization. *AIAA Journal*. 39(5), 877–884.

Schlichting, H., *et al.* 2004. *Boundary-layer theory*. Springer.

Smith, A.M., 1975. High-lift aerodynamics. *Journal of Aircraft*, 12 (6), 501–530.

Sóbester, A., 2009. Concise airfoil representation via case-based knowledge capture. *AIAA Journal*. 47 (5), 1209–1218.

Sobieszczanski-Sobieski, J., and Haftka, R. T., 1996. Multidisciplinary aerospace design optimization: survey of recent developments. *Structural Optimization*. 14, 1–23.

Sobieszczanski-Sobieski, J., and Venter, G., 2004. *Structural and Multidisciplinary Optimization*. 26, 121–131.

Sobjieczky, H., 1998. Parametric airfoils and wings. *Notes on Numerical Fluid Mechanics*. 68, 71–88.

Somers, D. M., 1981. Design and experimental results for a natural-laminar-flow airfoil for general aviation applications. *NASA Technical Paper-1861*.

Song, W., and Keane, A. J., 2004. A study of shape parameterisation methods for airfoil optimisation. In *Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. Albany, New York.

Sobol, I. M., 1993. Sensitivity estimates for nonlinear mathematical models. *Mathematical Modeling and Computational Experiment*. 1(4), 407–414.

Sobol, I. M., 1994. A primer for the Monte Carlo method. CRC Press.

Spalart, P. R., and Allmaras, S. R., 1992. A one-equation turbulence model for aerodynamic flows. AIAA-Paper 92-0439. 1, 5–21.

Srinivas, N., and Deb, K., 1994. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*. 2, 221–248.

Storn, R., and Price, K., 1997. Differential evolution-A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*. 11, 341–359.

Thiele, L., Miettinen, P., Korhonen, P. J., and Molina, J., 2007. A preference-based interactive evolutionary algorithm for multobjective optimization. *Helsinki School of Economics*. W-412

Thompson, J. F., Thames, F. C., and Mastin C. W., 1974. Automatic numerical generation of body-fitted curvilinear coordinate system for fields containing any number of arbitrary two dimensional bodies. *Journal of Computational Physics*. 15(3), 299–319.

Thompson, P. A., 1986. *Compressible fluid dynamics*. McGraw-Hill.

Torczon, V. J., 1997. On the convergence of pattern search algorithms. *SIAM Journal on Optimization*. 7(1), 1–25.

Trelea, I. C., 2003. The particle swarm optimization algorithm: convergence analysis and parameter selection. *Information Processing Letters*. 85(6), 317-Ű325.

Trocine, L., and Malone, L., 2000. Finding important independent variables through screening designs. In *Proceedings of the 2000 Winter Simulation Conference*. Orlando, Florida.

Tudose, L., and Jucan, D., 2007. Pareto approach in multi-objective optimal design of helical compression springs. *Annals of the Oradea University*. 6(16).

van Dam, C.P., 2002. The aerodynamic design of multi-element high-lift systems for transport airplanes. *Progress in Aerospace Sciences*, 38 (2), 101–144.

Vatsa, V.N., *et al.*, 1994. Multi-block structured grid approach for solving flows over complex aerodynamic configurations. AIAA Paper 94-0655.

Vavelle, A., and Qin, N., 2007. Iterative response surface based optimization scheme for transonic airfoil design. *Journal of Aircraft*. 44(2), 365–376.

Vicini, A., and Quagliarella, D., 1997. Inverse and direct airfoil design using a multiobjective genetic algorithm. *AIAA Journal*. 35(9), 1499–1505.

Vicini, A., and Quagliarella, D., 1997. Airfoil and wing design through hybrid optimization strategies. *AIAA Journal*. 37(5), 634–341.

Wang, M., Lele, S. K., and Moin, P., 1996. Computation of quadrupole noise using acoustic analogy. *AIAA Journal*. 34(11), 2247Ű-2254.

Wang, X., and Damodaran, M., 2001. Aerodynamic shape optimization using computational fluid dynamics and parallel simulated annealing algorithms. *AIAA Journal*. 39(8), 1500–1508.

Wickramasinghe, U. K., and Li, X., 2008. Integrating user preferences with particle swarms for multi-objective optimization. In *Genetic and Evolutionary Computation Conference*. ACM Press. 745–752.

Wickramasinghe, U. K., and Li, X., 2009. Using a distance metric to guide PSO algorithms for many-objective optimization. In *Genetic and Evolutionary Computation Conference*. ACM Press. 667–674.

Wickramasinghe, U. K., Carrese, R., and Li, X., 2010. Designing airfoils using a reference point based evolutionary many-objective particle swarm optimization algorithm. *IEEE World Congress on Computational Engineering*. Barcelona, Spain.

Wickramasinghe, U. K., *User-preference based evolutionary algorithms for many-objective optimisation*. PhD Dissertation. RMIT University. Melbourne, Australia.

Wierzbicki, A. P., 1980. The use of reference objectives in multiobjective optimization. *Multiple Criteria Decision Making Theory and Applications*. Springer-Verlag. 468Ű-486.

Yamamoto, K., and Inoue, O., 1995. Applications of genetic algorithm to aerodynamic shape optimization. In the *12th AIAA Computational Fluid Dynamics Conference*. San Diego, CA. 43–51.

Ying, S.X., Spaid, F.W., McGinley, C.B., and Rumsey, C.L., 1999. Investigation of confluent boundary layers in high-lift flows. *Journal of Aircraft*. 36 (3), 550–562.

Yiu, K. F. C., 1994. Computational methods for aerodynamic shape design. *Mathematical and Computer Modelling*. 20(12), 3–29.

Zingg, D. W., and Elias, S., 2006. Aerodynamic optimization under a range of operating conditions. *AIAA Journal*. 44(11), 2787–2792.

Zitzler, E., Deb, K., and Thiele, L., 2000. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*. 8, 173–195.